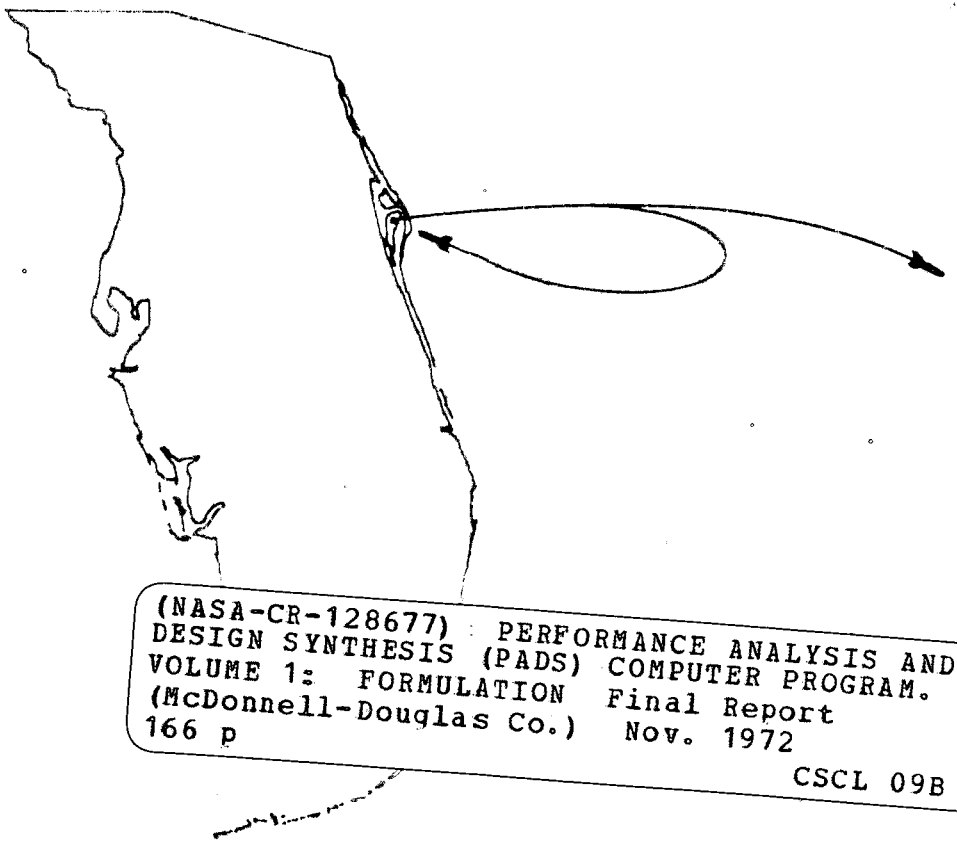
NOVEMBER 1972

MDC G4287



PERFORMANCE ANALYSIS AND DESIGN SYNTHESIS (PADS) COMPUTER PROGRAM

VOLUME I
Formulation
Final Report

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY

MCDONNELL DOUGLAS CORPORATION

MCDONNELL DOUGLAS

PERFORMANCE ANALYSIS AND DESIGN SYNTHESIS (PADS) COMPUTER PROGRAM

VOLUME I

Formulation

Final Report

NOVEMBER 1972

MDC G4287

PREPARED UNDER CONTRACT NO. NAS9-12059
BY GUIDANCE AND FLIGHT MECHANICS DEPARTMENT,
RESEARCH AND DEVELOPMENT
MCDONNELL DOUGLAS ASTRONAUTICS COMPANY—WEST
HUNTINGTON BEACH, CALIFORNIA
FOR
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY-WEST

5301 Bolsa Avenue, Huntington Beach, CA 92647

FOREWORD

This is the first of three volumes describing the Performance Analysis and Design Synthesis (PADS) computer program. This volume is devoted to a complete program formulation. Volume II contains programming and numerical techniques and Volume III is a user manual.

The development of PADS was conducted by McDonnell Douglas Astronautics Company at Huntington Beach, California, under NASA Contract NAS9-12059, under the cognizance of Mr. Robert Abel, NASA, MSC, Houston, Texas. The key MDAC personnel who formulated and programmed PADS are Messrs. Murray H. Rosenberg, John W. Hensley, and Michael Beach. Valuable programming assistance was given by Larry Ong, Fred Gangloff, and Sheldon Herman.

ABSTRACT

The Performance Analysis and Design Synthesis (PADS) computer program has a two-fold purpose. It can size launch vehicles in conjunction with calculus-of-variations optimal trajectories and can also be used as a general-purpose branched trajectory optimization program. In the former use, it has the Space Shuttle Synthesis Program as well as a simplified stage weight module for optimally sizing manned recoverable launch vehicles. For trajectory optimization alone or with sizing, PADS has two trajectory modules. The first trajectory module uses the method of steepest descent; the second employs the method of quasi-linearization, which requires a starting solution from the first trajectory module.

CONTENTS

TABLES

SYMBOLS

Symbols	Description	Unit
A	A matrix (steepest decent formulation)	
A_e	Net nozzle area	ft^2
A_{exit}	Nozzle area per engine	ft^2
a	Acceleration vector	
a^v	Component of a in $\dot{v}$ equation	
a^γ	Component of a in $\dot{\gamma}$ equation	
a^ψ	Component of a in $\dot{\psi}$ equation	
a^m	Component of a in $\dot{m}$ equation	
a_s	Semi-major axis	ft
a_o, a_1, a_2, a_3	Coefficient in booster-stage empty-weight equation	
B	B matrix	
b_o, b_1, b_2, b_3	Coefficients in orbiter-stage empty-weight equation	
C_D	Drag coefficient	
C_{D_o}	Zero-lift drag coefficient	
C_L	Lift coefficient	
C_{L_u}	Uncorrected lift coefficient	
C_{L_o}	Lift coefficient at $\alpha = 0$	
C_{L_α}	Lift coefficient slope	/rad
$C_{\mathscr{M}_o}$	Moment coefficient at $\alpha = 0$	
$C_{\mathscr{M}_\alpha}$	Moment coefficient slope	/rad
C_{u_L}	Constant lift term in governing equation	lb

SYMBOLS (Continued)

Symbols	Description	Unit
C_α	Constant angle of attack	rad
c_i	Multiplier for i^{th} homogeneous solution to linearized state and co-state equations	
D	Aerodynamic drag	lb
D ( )	Direction cosine matrix	
D'	Diagonal of A matrix	
D_b	Base drag	lb
D_y	Partial derivative operator including effect of steering vector u	
D^*_y	Partial derivative operator, steering vector u fixed	
d_{Ref}	Aerodynamic reference length	ft
$(dP)^2$	Metric of control and parameter changes	
E	Energy or eccentric anomaly	ft^2/sec^2 or rad
$E\{\delta u_i\}$	Unknown control functional	
E_R	Earth radius	ft
e	Orbit eccentricity	
F	Vector comprised of state and co-state differential equations in QL module	
F_{RATED}	Rated vacuum thrust (total)	lb
F_{VAC}	Vacuum thrust per engine	lb
f	Differential equations of motion	
$\bar{f}$	Nominal differential equations of motion	
G	Diagonal matrix of coefficients that multiplies acceleration vector	

SYMBOLS (Continued)

Symbols	Description	Unit
GM	Gravitational constant times Earth mass	
G_j	Variable part of cut-off function for arc j	
g	Gravitational acceleration	ft/sec^2
g_{max}	Maximum total acceleration	
g_r	Reference gravitational acceleration	ft/sec^2
H	Momentum (ft/sec), or Hamiltonian, or matrix of homogeneous solutions to linearized state and co-state equations	
h	Altitude	ft
h_i	i^{th} homogeneous solution to linearized state and co-state equations	
I_{sp}	Vacuum specific impulse	
I_{sp_B}	Booster vacuum I_{sp}	
$I_{sp_{eff}}$	Effective I_{sp} (with throttling)	
I_{sp_o}	Orbiter vacuum I_{sp}	
i	Inclination angle	
i_ρ	Unit vector along latitude	
i_μ	Unit vector along longitude	
J	Functional to be minimized	
j	Control blend factor	
K	Vector of governing equations with elements K_1, K_2, K_3	
K'	Ratio of dual engine I_{sp}'s	
k	Induced drag coefficient	
k_γ	Expression used in vertical rise or pitchover control governing equation	

SYMBOLS (Continued)

Symbols	Description	Unit
k_ψ	Expression used in vertical rise or pitchover control governing equation	
L	Aerodynamic lift	lb
L_u	Uncorrected lift	lb
L_{max}	Maximum lift	lb
M	Mach number, or minor of A matrix, or Lagrange multipliers for state boundary conditions	
$\mathcal{M}$	Aerodynamic moment	ft-lb
$\mathcal{M}_{CG}$	Aerodynamic moment about center of gravity	ft-lb
m	Mass	slugs
N	Minor vector of A matrix	
N_1	Last subarc of trunk	
N_2	Last subarc of first branch	
N_3	Last subarc of second branch ($N_2 = N_3$ if no branches)	
PL	Payload	lb
p	Adjustable parameter, or propulsion vector, or particular solution to linearized state and co-state equations	
p_a	Atmospheric pressure	lb/ft^2
Q	Heating	Btu/ft^2
$\dot{Q}$	Heating rate	$Btu/ft^2/sec$
Q_{MULT}	Heating flag	
q	Dynamic pressure (lb/ft^2) or control variables that are not free to be optimized	
R	Radial distance to vehicle	ft

SYMBOLS (Continued)

Symbols	Description	Unit
R'	Ratio of throttled dual engine thrust	
R_a	Apogee radius	ft
R_{ey}	Reynolds number (unit)	ft^{-1}
R_p	Perigee radius	ft
S_C	Cross-range	ft
S_D	Downrange	ft
SFC	Specific fuel consumption	lb/lb/hr
S_{Ref}	Aerodynamic reference area	ft^2
S_T	Total range	ft
S^{Ψ}	Matrix of adjustable parameter sensitivities	
s	Total solution to linearized state and co-state equations	
T	Thrust	lb
T_{MULT}	Number of engines	
Tol	Vector of constraint tolerances	
T_{VAC}	Total vacuum thrust	lb
t	Time (usually arc time)	sec
t_e	Elapsed time	sec
t_F	Trajectory end time	sec
t_o	Initial time	sec
$(\quad)^T$	Denotes transpose	
U	Control vector	
u	Steering vector	
V	Relative velocity	ft/sec

SYMBOLS (Continued)

Symbols	Description	Unit
V_A	Velocity at apogee	ft/sec
V_I	Inertial velocity	ft/sec
V_p	Velocity at perigee	ft/sec
W	Vehicle weight (lb), or control weighting matrix, or approximate solution to w	
W_{BO}	Booster burn-out weight	lb
W_{e_B}	Booster-stage empty weight	lb
W_{e_o}	Orbiter-stage empty weight	lb
$W_{L.O.}$	Lift-off weight	lb
W_o	Orbiter gross weight	lb
W_{p_B}	Booster propellant weight	lb
W_{p_o}	Orbiter propellant weight	lb
w	In-plane control vector (subset of U)	
X	Asymptotic injection parameter	ft
X_{CG}	Body x station instantaneous center-of-gravity	ft
X_{CGR}	Body x station reference center-of-gravity	ft
X_E	Body x station of engine thrust centroid	ft
X_T	Tail length body station	ft
x	Body station coordinate along axis (ft) or quasitime	
Y	Asymptotic injection parameter, or vector comprised of the QL module state and co-state vectors	
[Y]	Diagonal matrix of parameter weighting factors	
	State vector	

Section 1
INTRODUCTION

The Performance Analysis and Design Synthesis (PADS) computer program provides the capability to synthesize launch vehicle design in conjunction with an optimally shaped and staged trajectory. It also permits generalized trajectory optimization including branching for a wide variety of endo-atmospheric aerospace vehicles.

The synthesis capability, derived from the Space Shuttle Synthesis Program (SSSP), Reference 1, is oriented toward manned reusable launch vehicles. However, a simplified synthesis module is available in the program which provides a general two-stage launch vehicle design capability.

The trajectory and staging optimization portion of the program employs a closed-loop steepest descent method for an approximate solution. Moreover, the steepest descent solution may then be used as a starting guess for the program's quasi-linearization algorithm which determines the exact solution of the calculus of variations multipoint boundary value problem.

This document is the first of three volumes, and is devoted to the formulation of PADS. Volume II is the programmers document and Volume III is the User Manual. The bulk of Volume I is devoted to the development and discussion of the trajectory formulation. The first five sections describe the types of simulations that are available in the program, deferring the discussion of control and parameter optimization until later sections. Section 6 describes the general two-stage launch vehicle synthesis model (also called the Phase I sizing module). Section 7 is a brief discourse on the interplay between the trajectory model and the Space Shuttle Synthesis Program. The detailed documentation on this synthesis model is available in Reference 1.

Section 8 describes the auxiliary print computations that are available in the program.

In Sections 9 and 10 the general solution for the control vector is presented for non-optimal situations. This leads to the key variational aspects of the program, where optimal control and staging are formulated. Sections 11 through 15 show how the steepest-descent formulation calculates optimal steering and parameters. These sections include discussions of the adjoint differential equations, influence functions and numerical analysis and solution convergence techniques. Section 16 presents the details of the necessary conditions for the exact solution of the multi-point boundary value problem in preparation for a presentation in Section 17 of how the method of quasi-linearization satisfies these conditions.

Section 2

TRAJECTORY SIMULATION

In this section of the PADS formulation document, the earth and vehicle coordinate systems are defined and the equations of motion are given. In addition, the concept of the control vector, U, is defined.

2.1 EARTH-RELATIVE COORDINATE SYSTEM

The Earth-relative flight-path coordinate system is illustrated in Figure 2-1 below. The earth model is spherical-rotating with a central-force gravitational field. The origin of the R vector is geocentric.

2.2 VEHICLE COORDINATE SYSTEM

The vehicle is treated as a point-mass moving in three degrees of freedom. The applied load directions and control angles relative to the basic coordinate system are shown in Figure 2-2. It should be noted that the vehicle banks around the velocity vector, hence there is no yaw angle.

CR155-1

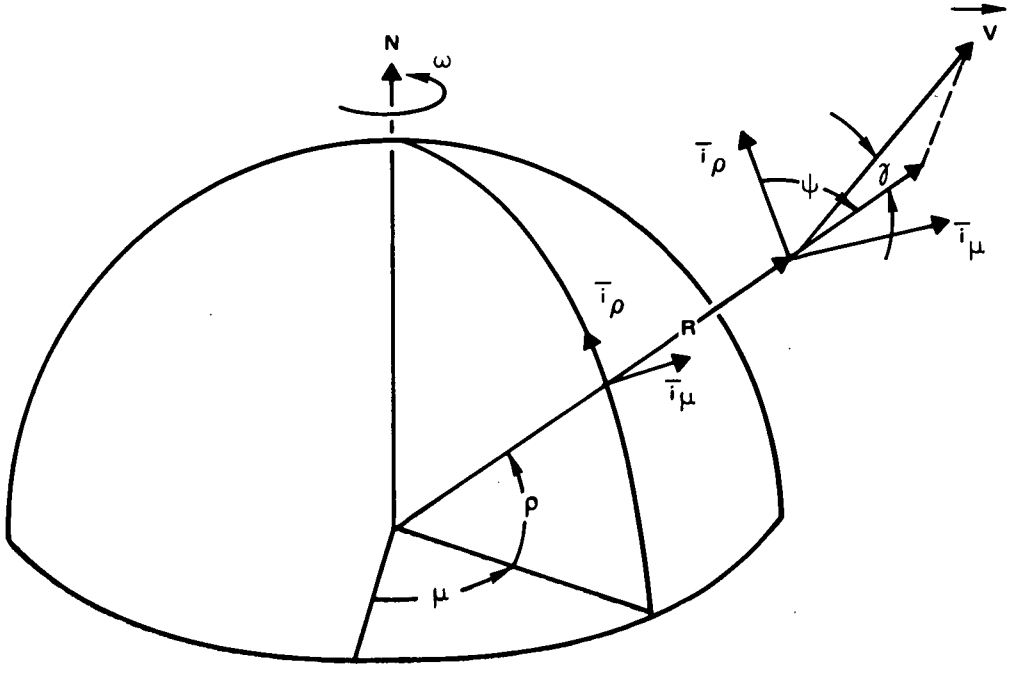


Figure 2-1. Earth-Relative Flight-Path Coordinate System

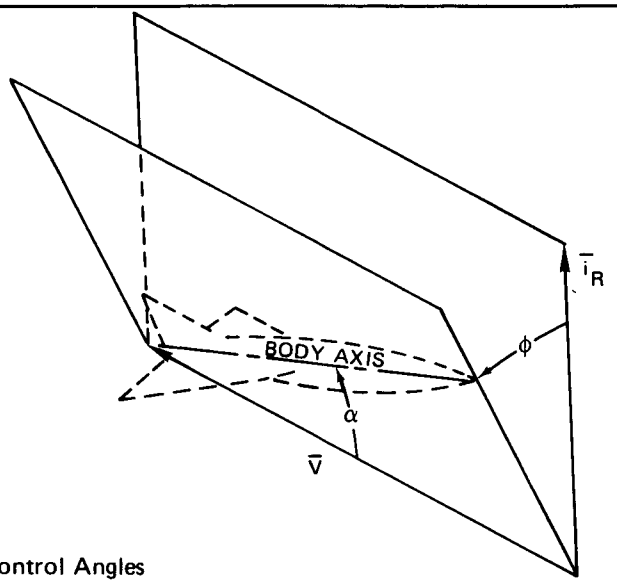


Figure 2-2. Vehicle Control Angles

2.3 EQUATIONS OF MOTION

The equations of motion in relative coordinates are derived in many other sources. They will be presented here in engineering notation. The meaning of various terms is given in Figures 2-1 and 2-2 and the acceleration components depend on the simulation model.

$$\dot{V} = R\omega^2 \cos\rho\,(\cos\rho \sin\gamma - \sin\rho \cos\psi \cos\gamma) - g \sin\gamma + \frac{a^V}{m} \qquad (2.3\text{-}1)$$

$$\dot{\gamma} = \omega \cos\rho\left[2 \sin\psi + \frac{R\omega}{V}(\cos\rho \cos\gamma + \sin\rho \cos\psi \sin\gamma)\right] + \cos\gamma\,(\frac{V}{R} - \frac{g}{V}) + \frac{a^\gamma}{V} \qquad (2.3\text{-}2)$$

$$\dot{\psi} = \frac{\omega\cos}{\cos\gamma}\left[\frac{R\omega \sin\rho \sin\psi}{V} - 2 \cos\psi \sin\gamma\right] + \sin\rho\left(\frac{V \cos\gamma \sin\psi}{R \cos\rho} + 2\omega\right) + \frac{a^\psi}{V \cos\gamma} \qquad (2.3\text{-}3)$$

$$\dot{R} = \dot{h} = V \sin\gamma \qquad (2.3\text{-}4)$$

$$\dot{\rho} = \frac{V}{R} \cos\gamma \cos\psi \qquad (2.3\text{-}5)$$

$$\dot{\mu} = \frac{V \cos\rho \sin\psi}{V \cos\rho} \qquad (2.3\text{-}6)$$

The rate of change of mass is

$$\dot{m} = a^{m} \qquad (2.3\text{-}7)$$

The equation for heating rate is

$$\dot{Q} = Q_{MULT} \left[17600. \sqrt{\sigma}^{\ 1} \left(\frac{V}{26000.}\right)^{3.15}\right] \qquad (2.3\text{-}8)$$

Particular terms in the above equations are defined below by equations or functional dependencies.

$$g = \frac{GM}{R^2} \qquad (2.3\text{-}9)$$
(EQUA3, STATEF)

The following functional dependencies are characteristic of the type of aerodynamic and propulsion simulations. *

LIFT: $L = qS_{Ref} C_L \{\alpha, M\}$ (2.3-10)
(VT, UT)

DRAG: $D = qS_{Ref} C_D \{\alpha, M\}$ (2.3-11)
(VT, UT)

BASE DRAG $D_b = D_b \{h\}$ (2.3-12)
(EQUA3, STATEF)

We have purposely omitted the functional dependencies of T, δ_E, α and ϕ and a, which will be discussed in Section 2.4.

*T, L, and D are model-dependent force terms described in Section 3.

2.4 INTRODUCTION TO THE CONTROL VECTOR

If we examine the equations of motion we find that they contain kinematic and dynamic terms. The dynamic terms involve the applied loads. In a general sense, the equations of motion may be expressed as

$$\dot{y} = f\{y\} + G\,[a] \tag{2.4-1}$$

where, a, called the acceleration vector, may be defined for example as*

$$a = \begin{pmatrix} a_V \\ a_\gamma \\ a_\psi \\ a_m \end{pmatrix} = \begin{bmatrix} \dfrac{T\cos(\alpha - \delta_E) - D - D_b \cos\alpha}{m} \\ \dfrac{T\sin(\alpha - \delta_E) + L - D_b \sin\alpha}{m} \quad \cos\phi \\ \dfrac{T\sin(\alpha - \delta_E) + L - D_b \sin\alpha}{m} \quad \sin\phi \\ \dot{m} \end{bmatrix} \tag{2.4-2}$$

(ACCEL, APPLY VT, UT)

and G is a diagonal matrix whose elements are

$$[G] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{1}{V} & 0 & 0 \\ 0 & 0 & \dfrac{1}{V\cos\gamma} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4-3}$$

The functional dependency of "a" may be expressed

$$a = a\{T,\ \delta_E,\ \alpha,\ \phi,\ y,\ t\} \tag{2.4-4}$$

*The example given is for the single-engine moment balance simulation.

Of the dependencies exhibited in Equation (2.4-4), it is apparent that the "decision quantities" are T, δ_E, α, and ϕ.

We group these into the vector U, called the control vector

$$U^T = (T, \delta_E, \alpha, \phi) \tag{2.4-5}$$

and using this notation rewrite Equation (2.4-1)

$$\dot{y} = f\{y\} + [G]\, a\{U, y, t\} \tag{2.4-6}$$
(DER3A, NLDRV)

For convenience, in later discussion, the control vector may be divided into subsets. We class T, δ_E, and α as the in-plane control vector, w, and define the steering vector as:

$$u^T = (\alpha, \phi)$$

As has been described in Section 1, the purpose of the trajectory optimization portions of PADS is to solve for the time history of U that satisfies all algebraic and variational problem constraints.

Section 3
APPLIED LOADS MODELS

This section describes the various applied loads models currently available in PADS. The applied loads are divided into aerodynamic and propulsion models.

3.1 AERODYNAMIC MODELS

There are three types of aerodynamic models available in PADS. These are: (1) asymmetric linear lift variation with quadratic drag polar; (2) bivariate tabular lift and drag coefficients as functions of angle of attack and Mach number; and (3) static moment balance aerodynamics.

3.1.1 Asymmetric Linear Aerodynamic Model

The equations for total lift and drag coefficients depend on coefficients that are input tabular functions of Mach number.

$$C_L = C_{L_\alpha}\, \alpha + C_{L_O} \qquad (3.1\text{-}1)$$
(BERØCØ, AERØCØ)

$$C_D = C_{D_O} + k\, C_L^{\,2} \qquad (3.1\text{-}2)$$
(BERØCØ, AERØCØ)

The aerodynamic lift and drag are calculated from these coefficients as:

$$L = C_L\, q\, S_{REF} \qquad (3.1\text{-}3)$$
(VT, UT)

$$D = C_D\, q\, S_{REF} \qquad (3.1\text{-}4)$$
(VT, UT)

where

$$q = 1/2\ \rho\ V^2 \qquad (3.1\text{-}5)$$

(EQUA3, STATEF)

3.1.2 Nonlinear Aerodynamics Model

The nonlinear aerodynamics model employs bivariate tables of lift and drag coefficients as functions of angle of attack and Mach number. These tables are fitted with bicubic spline functions to yield continuous first, second, and mixed partial derivatives during program execution. (BLYNE, BLICO, BLINE)

3.1.3 Base Drag

Base drag, D_b, in the program is a function of altitude. It is an input univariant table.

3.1.4 Moment Balance Aerodynamics Model

The moment balance aerodynamics model includes the effect of aerodynamic and propulsive moment balance trim forces on the overall applied loads. The distribution of trim between aerodynamic and propulsive is accounted for through a blend factor, j. The moment balance diagram for this model is given in Figure 3.1-1.

Asymmetric linear aerodynamics are employed with the addition of aerodynamic moments.

The uncorrected lift (untrimmed lift) is

$$C_{L_u} = \left(C_{L_\alpha}\ \alpha + C_{L_O} \right) \qquad (3.1\text{-}6)$$

(AEROCO, BEROCO)

$$L_u = C_{L_u}\ q\ S_{Ref} \qquad (3.1\text{-}7)$$

(VT, UT)

The equation for drag is

$$C_D = C_{D_o} + k \left(C_{L_u} \right)^2 \qquad (3.1\text{-}8)$$

(AEROCO, BEROCO)

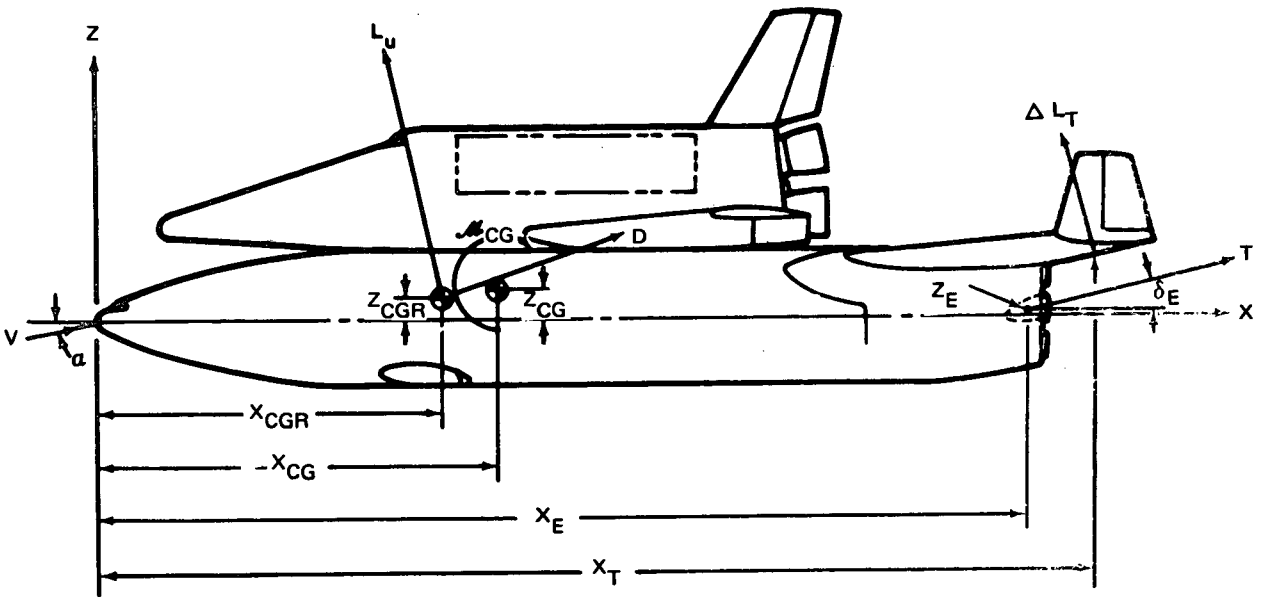


Figure 3.1-1. Moment Balance Diagram—Body Station Coordinate System

$$D = C_D \, q \, S_{Ref} \qquad (3.1\text{-}9)$$
(VT, UT)

The aerodynamic moment calculation assumes linear variation of moment coefficient with angle of attack.

$$\mathcal{M} = \left(C_{\mathcal{M}_\alpha} \, \alpha + C_{\mathcal{M}_o} \right) q \, S_{Ref} \, d_{Ref} \qquad (3.1\text{-}10)$$
(MOMECO, MAMECO UT, VT)

The moment coefficients $C_{\mathcal{M}_\alpha}$ and $C_{\mathcal{M}_o}$ are defined for moments about the reference center-of-gravity location (X_{CGR}, Z_{CGR}) only. The actual center-of-gravity location is an input tabular function of vehicle weight.

$$X_{CG} = X_{CG}(W) \text{ and } Z_{CG} = Z_{CG}(W) \qquad (3.1\text{-}11)$$
(EQUA3, STATEF)

In developing the equation for the total aerodynamic moment around the instantaneous vehicle center of gravity, the following assumptions are made.

A. The base drag acts parallel to the vehicle axis and is centered at station Z_{BD}.

B. The axial contribution of aerodynamic trim force may be neglected. This assumption is very good for aerodynamically stable airframes with trim surfaces aft of the center of gravity. The assumption is poor for moderate to highly unstable or extremely stable airframes since in either case, induced drag changes due to trim deflection should then be accounted for.

C. The aerodynamic lift coefficient slope, C_{L_α}, should be fitted to vehicle data without aerodynamic trim surface deflection (untrimmed).

The untrimmed moment about the center of gravity is (approximately)

$$\mathcal{M}_{CG} = (L_u \cos\alpha + D \sin\alpha)(X_{CG} - X_{CGR}) - (D \cos\alpha - L_u \sin\alpha)(Z_{CG} - Z_{CGR}) + \left(C_{\mathcal{M}_o} + C_{\mathcal{M}_\alpha}\alpha\right) q S_{Ref} d_{Ref} + D_B(Z_{BD} - Z_{CG}) \quad (3.1\text{-}12)$$

(VT, UT)

To distribute the trim force between the aerodynamic trim device (tail) and the engine thrust gimbal contribution, the blend factor, j, is used. j is an input function of dynamic pressure, q.

$$j = j(q) \quad (3.1\text{-}13)$$

(EQUA3, STATEF)

The tail contribution then is

$$(X_T - X_{CG})\,\Delta L_T = j\mathcal{M}_{CG} \quad (3.1\text{-}14)$$

or

$$\Delta L_T = \frac{j\mathcal{M}_{CG}}{(X_T - X_{CG})} \qquad (3.1\text{-}15)$$
(VT, UT)

The corrected or total lift then becomes

$$L = L_u + \Delta L_T \qquad (3.1\text{-}16)$$
(VT, UT)

The engine thrust contribution is

$$\left\{(X_E - X_{CG}) \sin \delta_E - (Z_E - Z_{CG}) \cos \delta_E \right\} T = (1 - j)\mathcal{M}_{CG} \qquad (3.1\text{-}17)$$

The equations above are employed in developing the governing equation set, K, for the in-plane control vector, w. The solution procedure is described in Section 9.2.

3.2 PROPULSION MODELS

Four propulsion models are employed in PADS. The first is a simple rocket model using input vacuum thrust with optional throttling and computing fuel flow. The second, for use with the SSSP sizing module, simulates two engines with different I_{sp}'s, with optional throttling. The third model is an air breather simulation. The fourth model has dual parallel-burn engines.

3.2.1 Simple Rocket Model

The rocket vacuum thrust per engine, F_{VAC}, may be input as a tabular function of burn time or as a constant. The net thrust of the vehicle may then be calculated as:

$$T = \left[F_{VAC} - A_{EXIT} P_a\right] \times T_{MULT} \qquad (3.2\text{-}1)$$
(EQUA3, STATEF)

where

A_{EXIT} is the nozzle exit area,

p_a is ambient static pressure,

T_{MULT} is the number of engines or thrust multiplier.

The vehicle rate of change of mass may then be calculated.

$$\dot{m} = - \frac{F_{VAC}\, T_{MULT}}{I_{sp}\, g_r} \qquad (3.2\text{-}2)$$
(ACCEL, APPLY)

When acceleration limit throttling is employed, the net thrust, T, is computed using an appropriate governing equation (K) so that the total vehicle acceleration is bounded. The total vacuum thrust may then be calculated

$$T_{VAC} = T + A_{EXIT}\, p_a\, T_{MULT} \qquad (3.2\text{-}3)$$
(TH, FH)

If no I_{sp} loss table has been input, the vehicle rate of change of mass is then

$$\dot{m} = - \frac{T_{VAC}}{I_{sp}\, g_r} \qquad (3.2\text{-}4)$$
(ACCEL, APPLY)

If on the other hand, an I_{sp} loss table

$$\%I_{sp} = \%I_{sp} \left\{ \left(\frac{T_{VAC}}{F_{RATED}} \right) x100 \right\} x\ .01 \qquad (3.2\text{-}5)$$
(IMPUL, IMPULS)

has been input, the effective $I_{sp_{eff}}$ will be used and

$$I_{sp_{eff}} = I_{sp} \times \%ISP \left(\frac{T_{VAC}}{F_{rated}} \right) \times .01 \qquad (3.2\text{-}6)$$

(IMPUL, IMPULS)

and

$$\dot{m} = - \frac{T_{VAC}}{I_{sp_{eff}} g_r} \qquad (3.2\text{-}7)$$

(ACCEL, APPLY)

where F_{rated} is either input as a constant or interpolated from the input thrust table at the initiation of the thrusting (initial burn time).

3.2.2 Dual Engine Model

The dual engine model used with SSSP sizing problems first calculates the effective I_{sp} of the two engines with T_{VAC_1} and T_{VAC_2}, their rated vacuum thrust levels.

$$T_{VAC} = T_{VAC_1} + T_{VAC_2} \qquad (3.2\text{-}8)$$

$$I_{sp} = \frac{I_{sp_1} I_{sp_2} \left(T_{VAC_1} + T_{VAC_2} \right)}{T_{VAC_1} I_{sp_2} + T_{VAC_2} I_{sp_1}} \qquad (3.2\text{-}9)$$

(ISPRAT)

This model assumes that only the second engine may have time-varying thrust (table input) or may be throttled. The net effective I_{sp} for the system will change if this is permitted. We define A_e as the net nozzle area and given the net instantaneous thrust T, we may calculate the ratio of actual vacuum to rated vacuum thrust of the two engines.

$$R' = \frac{T + A_e p_a}{T_{VAC_1} + T_{VAC_2}} \qquad (3.2\text{-}10)$$
(ISPRAT)

We also define

$$K' = \frac{I_{sp_2}}{I_{sp_1}} \qquad (3.2\text{-}11)$$
(ISPRAT)

We may then calculate the ratio of instantaneous I_{sp}' to that defined in Equation (3.2-9).

$$\frac{I_{sp}'}{I_{sp}} = R' \left[\frac{K' T_{VAC_1} + T_{VAC_2}}{(K' + R' - 1) T_{VAC_1} + R' T_{VAC_2}} \right] \qquad (3.2\text{-}12)$$
(ISPRAT)

The equation for effective I_{sp} and its partial derivatives are programmed in subroutine ISPRAT. The program logic is designed to parallel the I_{sp} loss table results by using the same type of dependencies; that is, the result of both calculations is the percent I_{sp} as a function of percent rated thrust.

3.2.3 Air-Breather Propulsion

The air-breather propulsion model employs bivariate tables of thrust, T, and specific fuel consumption, SFC, as functions of relative velocity, V, and altitude, h. The SFC is in the units of fuel per lb-thrust per hour. The rate of change of vehicle mass may be calculated as follows:

$$\dot{m} = -\frac{(SFC)\, T}{g_r\ 3600} \qquad (3.2\text{-}13)$$
(ACCEL, APPLY)

3.2.4 Parallel Burn Propulsion Model, JPRØ = 3, JAER = 3

The configuration of the parallel burn model is shown in Figure 3.2-1. In the equations, all aerodynamic coefficients are functions of Mach Number, M.

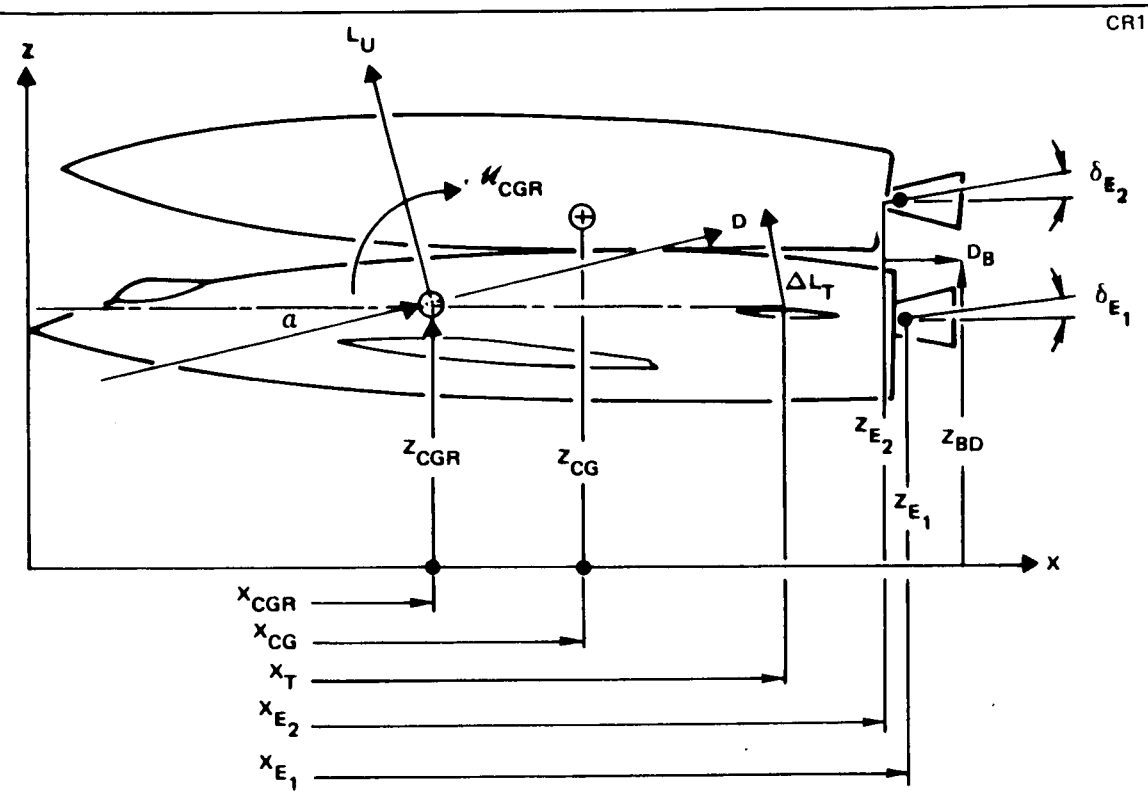


Figure 3.2-1. Parallel Burn Configuration

Aerodynamic lift (untrimmed):

$$L_u = \left(C_{L_\alpha}\,\alpha + C_{L_o}\right) q\, S_{Ref} \qquad (3.2\text{-}14)$$

Drag coefficient:

$$C_D = C_{D_o} + k\left(C_{L_u}\right)^2 \qquad (3.2\text{-}15)$$

Aerodynamic moment:

$$\mathcal{M}_a = \left(C_{\mathcal{M}_\alpha}\,\alpha + C_{\mathcal{M}_o}\right) q\, S_{Ref}\, d_{Ref} \qquad (3.2\text{-}16)$$

Center-of-gravity dependency:

$$Z_{CG'} = Z_{CG}\,(W) \qquad (3.2\text{-}17)$$

where W is vehicle total weight

$$X_{CG} = X_{CG}(W) \tag{3.2-18}$$

For this model, base drag is centered at station Z_{BD}.

The total untrimmed moment, $\mathcal{M}_{CG}$, about the instantaneous center of gravity depends on which engine is being gimballed or whether both engines are gimballed. Assume the β engine, having thrust T_β, is not gimballed but has fixed engine deflection, δ_{E_β}. Then

$$\begin{aligned}\mathcal{M}_{CG} = {} & \left(L_u \cos\alpha + D \sin\alpha\right)\left(X_{CG} - X_{CGR}\right) \\ & - \left(D\cos\alpha - L_u \sin\alpha\right)\left(Z_{CG} - Z_{CGR}\right) + \mathcal{M}_a + D_B\left(Z_{BD} - Z_{CG}\right) \\ & - \left(Z_E - Z_{CG}\right) T_\beta \cos\delta_{E_\beta} + \left(X_E - X_{CG}\right) T_\beta \sin\delta_{E_\beta}\end{aligned} \tag{3.2-19}$$

If both engines are gimballed, the last two terms involving T_β are excluded.

The tail contribution to balance the moment depends on the blend factor $j = j(q)$, where q is the dynamic pressure $\left(= \frac{\rho}{2} V^2\right)$

$$\Delta L_T = \frac{j\,\mathcal{M}_{CG}}{\left(X_T - X_{CG}\right)} \tag{3.2-20}$$

For one-engine fixed, the gimballable engine deflection for engine γ required to balance the remainder of the moment is:

$$\left[\left(X_{E_\gamma} - X_{CG}\right)\sin\delta_{E_\gamma} - \left(Z_{E_\gamma} - Z_{CG}\right)\cos\delta_{E_\gamma}\right] T_\gamma = (1-j)\,\mathcal{M}_{CG} \tag{3.2-21}$$

If both engines are gimballed, we have

$$\begin{aligned}&\left[T_\beta\left(X_{E_\beta} - X_{CG}\right) + T_\gamma\left(X_{E_\gamma} - X_{CG}\right)\right]\sin\delta_E \\ &- \left[T_\beta\left(Z_{E_\beta} - Z_{CG}\right) + T_\gamma\left(Z_{E_\gamma} - Z_{CG}\right)\right]\cos\delta_E = (1-j)\,\mathcal{M}_{CG}\end{aligned} \tag{3.2-22}$$

This completes the governing equation for engine-deflection solution. It remains to describe the applied-load terms in the equation of motion. The general form is

$$\dot{y} = f(y) + Ga \tag{3.2-23}$$

where f(y) are the applied-load independent terms, G is a diagonal matrix of applied-load independent multiplying factors, and a is the acceleration vector having elements

$$a = \begin{pmatrix} a^V \\ a^\gamma \\ a^\psi \\ a^m \end{pmatrix} \tag{3.2-24}$$

and

$$\text{diag } [G] = \left[1, \frac{1}{V}, \frac{1}{v \cos\gamma}, \frac{1}{g_r}\right] \tag{3.2-25}$$

The equations for the first three elements of a are

$$a^V = \frac{T_1 \cos\left(\alpha + \delta_{E_1}\right) + T_2 \cos\left(\alpha + \delta_{E_2}\right) - D - D_B \cos\alpha}{m} \tag{3.2-26}$$

let

$$a^X = \frac{T_1 \sin\left(\alpha + \delta_{E_1}\right) + T_2 \sin\left(\alpha + \delta_{E_2}\right) + L - D_B \sin\alpha}{m} \tag{3.2-27}$$

then

$$a^\gamma = a^X \cos\phi \tag{3.2-28}$$

$$a^\psi = a^X \sin\phi \tag{3.2-29}$$

The equation for a^m depends on T_1 and T_2. (Note: T_2 may not be throttled)

$$a^m = -\frac{T_1}{ISP_1} - \frac{T_2}{ISP_2} \qquad (3.2\text{-}30)$$

where

T_1 and T_2 are functions of t

and

ISP_1 and ISP_2 are constants.

3.3 ATMOSPHERIC MODELS

Three atmospheric models are available in PADS:

A. 1962 standard

B. 1963 Patrick AFB

C. Vacuum.

3.3.1 1962 Standard Atmosphere

The 1962 Standard Atmosphere model is a coded version of the analytic representation of Reference 2 up to 195 km. Above 195 km, a 50-point weighted least squares polynomial extends the range of data to beyond the altitudes of interest. This model is contained in subroutines ANLATM in the steepest descent portion of the trajectory module and in ANL62S in the quasi-linearization portion.

3.3.2 1963 Patrick AFB Model

The 1963 Patrick AFB model is described in Reference 3. It employs 14th-order polynomials for atmospheric properties up to 400,000 ft.

Section 4
AERODYNAMIC HEATING MODEL

The aerodynamic heating model employed in PADS calculates the stagnation heating rate on a spherical nose cap. The equation is

$$\dot{Q} = Q_{MULT}\left[\frac{17600}{\sqrt{\text{nose radius}}}\left(\frac{\rho_a}{\rho_b}\right)^{1/2}\left(\frac{V}{26000}\right)^{3.15}\right] \quad (4\text{-}1)$$

(DER3A, NLDRV)

where the nose radius is assumed to be 1 and ρ_b is the base density of the atmosphere model. Q_{MULT} is an input flag having a value of either 1. or zero depending on whether the heat load should be calculated in that portion of the trajectory. Q_{MULT} is necessarily an arc-dependent flag.

Section 5
MISSION CAPABILITIES

The mission capabilities of the trajectory program are described in this section. These mission capabilities are, in a mathematical sense, the boundary conditions on the multi-point boundary value problem. By way of introduction to these boundary conditions, some definitions will be useful in later discussions of the calculus-of-variations formulation (see Section 17).

There are two types of boundary conditions, initial conditions and targets. Initial conditions apply to the beginning of an arc whereas targets occur only at the end of an arc. Several types of initial conditions can occur in a problem. These are

A. Fixed initial condition

$$y_i = (\text{known value}) \tag{5.0-1}$$

where y_i is element of the state vector.

B. Continuous initial condition

$$y_i \big|_{\tau+} = y_i \big|_{\tau-} \tag{5.0-2}$$

where $|_\tau$ denotes the arc end point

C. Known or computable mass discontinuity

$$m \big|_{\tau+} = m \big|_{\tau-} - \Delta m \tag{5.0-3}$$

D. Mass distribution

Let superscript 1 denote branch one and superscript 2 denote branch two. Then

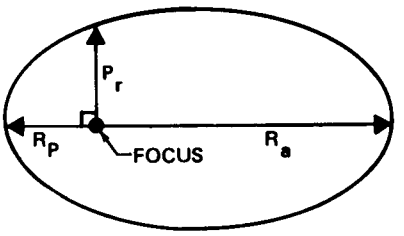
$$m \Big|^{1}_{\tau+} + m \Big|^{2}_{\tau+} = m \big|_{\tau-} \tag{5.0-4}$$

The target conditions that can be met are described in the remainder of this section. These targets are divided into ascent, entry, or auxiliary boundary conditions. In addition to those described, it should be noted that the relative state vector is also a set of candidate targets. It should also be noted that all targets described are functions of the relative state and time only.

5.1 ASCENT BOUNDARY CONDITIONS

The orbital parameters are illustrated in Figure 5-1, programmed in PDBC and PDBCQL, and listed below.

CR155-I



PLANAR ELLIPTICAL

P_r SEMI-LATUS RECTUM

R_P PERIGEE

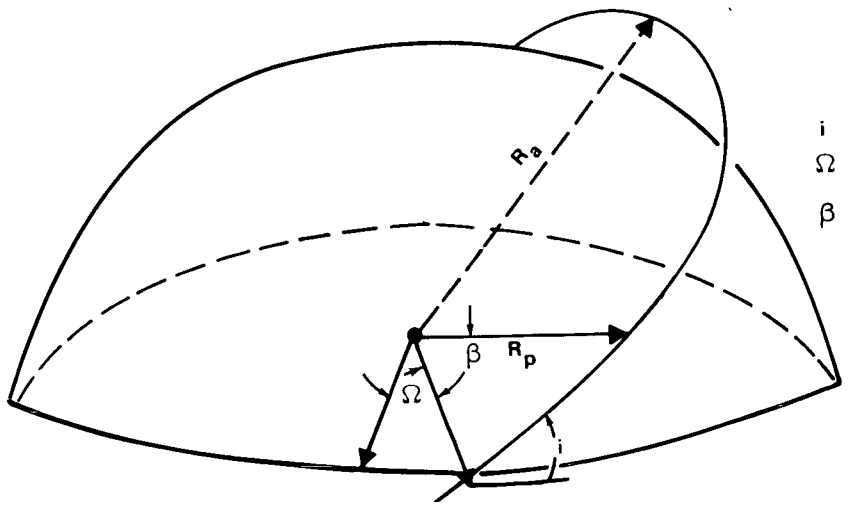
R_a APOGEE

$a_S = \frac{R_P + R_a}{2}$ SEMI-MAJOR AXIS

$e = \frac{R_a - R_P}{R_a + R_P}$ ECCENTRICITY

$H = \sqrt{GMp}$

$E = 2GM/a$, TRUE ANOMALY $= \text{TAN}^{-1}\left(\frac{\text{TAN}\,\gamma_I}{1 - \frac{r}{p}}\right)$



OUT-OF PLANE

i INCLINATION

Ω LONGITUDE OF ASCENDING NODE

β ARGUMENT OF PERIGEE

Figure 5-1. Orbital Parameters

A. Inertial velocity

$$V_I = \sqrt{V^2 + 2\omega\cos\rho\, V\cos\gamma\sin\psi + (\omega\cos\rho)^2}$$

B. Inertial path angle

$$\gamma_I = \sin^{-1}(V\sin\gamma / V_I)$$

C. Inertial azimuth

$$\psi_I = \tan^{-1}\left(\frac{V\cos\gamma\sin\psi + \omega\cos\rho}{V\cos\gamma\cos\psi}\right)$$

D. Inertial longitude

$$\mu_I = \mu + \omega t_e$$

E. Semi-latus rectum

$$P_r = \frac{R^2 V_I^2 \cos^2\gamma}{GM}$$

F. Orbital eccentricity

Let

$$Z = \frac{RV_I^2}{GM}$$

$$e = \sqrt{1 - Z(2-Z)\cos^2\gamma_I}$$

G. Orbital inclination

$$i = \cos^{-1}(\cos\rho\sin\psi_I)$$

H. Argument of perigee

$$\beta_p = \sin^{-1}\left(\frac{\sin\rho}{\sin i}\right) - \cos^{-1}\left(\frac{p_r - R}{E_R}\right)$$

I. Longitude of the ascending node

$$\Omega = \mu_I - \sin^{-1}\left(\frac{\sin\psi_I \sin\rho}{\sin i}\right)$$

J. Semi-major axis

$$a_s = \frac{R\,GM}{2\,GM - RV_I^2}$$

K. Apogee radius

$$R_a = a_s\,(1 + e)$$

L. Perigee radius

$$R_p = a_s\,(1 - e)$$

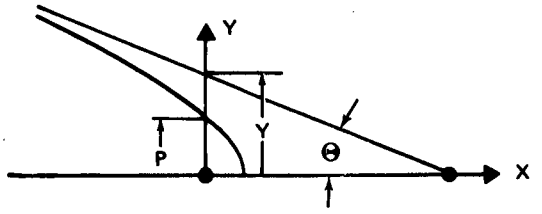
M. True anomaly

$$\zeta = \tan^{-1}\left(\frac{\tan\gamma_I}{1 - \frac{R}{p_r}}\right)$$

The following three parameters are applicable to asymptotic injection. The parameters are illustrated in Figure 5-2.

A. $\overline{X} = \frac{p_r}{e}$

B. $\overline{Y} = \frac{p_r e}{\sqrt{1-e^2}}$



Θ ASYMPTOTE
Y NORMAL DISTANCE
X HYPERBOLA AXIS DISTANCE

Figure 5-2. Asymptotic Parameters

C. $\Theta = \cos^{-1}\left(\frac{1}{e}\right)$

Two additional orbital parameters are energy and momentum.

A. Energy

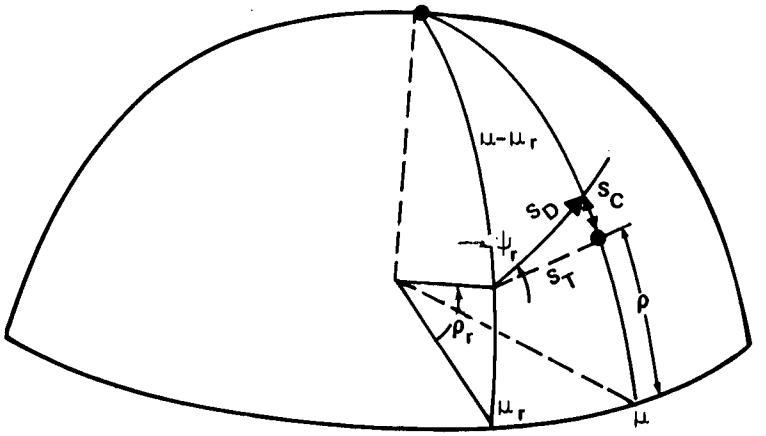
$$E = \frac{2\ GM}{a_s}$$

B. Momentum

$$H = R\ V_I \cos \gamma_I$$

5.2 ATMOSPHERE ENTRY MISSIONS

Figure 5-3 illustrates the how total range, downrange, and cross-range are defined.



μ_r = REFERENCE LONGITUDE
ρ_r = REFERENCE LATITUDE
ψ_r = REFERENCE AZIMUTH
μ = VEHICLE LONGITUDE
ρ = VEHICLE LATITUDE
S_D = DOWNRANGE
S_C = CROSS-RANGE
S_T = TOTAL RANGE

Figure 5-3. Range Targets

The equations for these are given below:

A. Downrange

$$S_D = E_R \tan^{-1}\left[\frac{\cos\rho_r(\sin\rho\cos\rho_r - \cos\rho\sin\rho_r\cos\Delta\mu) + \sin\psi_r\cos\rho\sin\Delta\mu}{\sin\rho\sin\rho_r + \cos\rho\cos\rho_r\cos\Delta\mu}\right]$$

where $\Delta\mu = \mu - \mu_r$ and the 'r' subscript means reference point.

B. Cross-range

$$S_C = E_R \sin^{-1}\left[\cos\psi_r\cos\rho\sin\Delta\mu - \sin\psi_r(\sin\rho\cos\rho_r - \cos\rho\sin\rho_r\cos\Delta\mu)\right]$$

C. Total range

$$S_T = E_R \cos^{-1}\left[\sin\rho\sin\rho_r + \cos\rho\cos\rho_r\cos\Delta\mu\right]$$

5.3 AUXILIARY BOUNDARY CONDITIONS

Additional boundary conditions in PADS for various purposes are given below.

A. Dynamic pressure

$$q = \frac{\rho_a V^2}{2}$$

B. Heating rate

$$\dot{Q} = Q_{MULT}\left[17600\sqrt{\sigma}\left(\frac{V}{26000}\right)^{3.15}\right]$$

C. Reynolds number (unit)

$$R_{ey} = \frac{V}{\nu}$$

The payload boundary condition equations are described in Sections 6 and 13.

Section 6

PHASE I SIZING

There are six sizing options in the Phase 1 sizing. These options are:

1. Maximize payload or burnout weight with fixed initial weight
2. Minimize lift-off weight with a fixed payload
3. Minimize lift-off weight with a fixed payload and second stage
4. Minimize lift-off weight with a fixed payload and first stage
5. Maximize payload or burnout weight with a fixed $(T/W)_{L.O.}$ and thrust
6. Maximize payload or burnout weight with a fixed $(T/W)_{L.O.}$ and initial weight

Each option is solved in its own subroutine and discussed below.

6.1 OPTION 1

The equations for sizing option 1 are based on the initial weight and the mass ratios that come from the trajectory program. The booster propellant weight is given by

$$W_{p_B} = W_{L.O.}\left(\frac{\mu_B - 1}{\mu_B}\right) \tag{6.1-1}$$

and the burnout weight by

$$W_{BO} = W_{L.O.} - W_{p_B} \tag{6.1-2}$$

The orbiter inert weight is determined from either a tabular input of the booster stage weight or from the following expression for this parameter

$$W_{e_B} = a_o + a_1 w_{p_B} + a_2 w_{p_B}^{1/3} + a_3 w_{p_B}^{2/3} \tag{6.1-3}$$

or

$$W_{e_b} = f(W_{p_B})$$

Then,

$$W_o = W_{BO} - W_{e_B} \tag{6.1-4}$$

The rest of the stage parameters are known.

6.2 OPTION 2

The second option requires the sizing to be done from the top down. The orbiter mass ratio is determined from

$$\mu_o = \exp\left(\frac{\Delta V_{TOT} - g_r\, Isp_B\, \ell n(\mu_B)}{g_r\, I_{sp_o}}\right) \tag{6.2-1}$$

and the propellant weight is given by

$$W_{p_o} = \left(\mu_o - 1\right)\left(PL + W_{e_o}\right) \tag{6.2-2}$$

where

$$W_{e_o} = b_o + b_1\, W_{p_o} + b_2\, W_{p_o}^{1/3} + b_3\, W_{P_o}^{2/3} \tag{6.2-3}$$

or is input via tabular data.

The orbiter initial weight is given by

$$W_o = W_{p_o} + W_{e_o} + PL \tag{6.2-4}$$

The booster weights are determined from

$$W_{p_B} = \left(\mu_B - 1\right)\left(W_o + W_{e_B}\right) \tag{6.2-5}$$

$$W_{BO} = W_o + W_{e_B} \tag{6.2-6}$$

and W_{e_B} is determined from Equation (6.1-3) or tabular input. The lift-off weight is given by

$$W_{L.O.} = \mu_B \, W_{BO} \tag{6.2-7}$$

6.3 OPTION 3

The option 3 sizing starts with the determination of the orbiter gross weight

$$W_o = W_{p_o} + W_{e_o} + PL \tag{6.3-1}$$

These quantities are input.

The booster mass ratio is given by

$$\mu_B = \text{EXP}\left(\frac{\Delta V_{TOT} - g_r I_{sp} \, \ell n \left(\dfrac{W_o}{PL + W_{e_o}}\right)}{g_r I_{sp_B}}\right) \tag{6.3-2}$$

The propellant weight is given by

$$W_{p_B} = \left(\mu_B - 1\right)\left(W_o + W_{e_B}\right) \tag{6.3-3}$$

where W_{e_B} is given by Equation (6.1-3). The booster burnout weight is determined from

$$W_{BO} = W_o + W_{e_B} \tag{6.3-4}$$

and the booster lift-off weight by

$$W_{L.O.} = \mu_B \, W_{BO} \tag{6.3-5}$$

6.4 OPTION 4

This sizing option requires the minimization of the lift-off weight. The booster lift-off weight is given by

$$W_{L.O.} = W_{p_B} \left(\frac{\mu_B}{\mu_B - 1} \right) \tag{6.4-1}$$

and the burnout weight by

$$W_{BO} = \frac{W_{L.O.}}{\mu_B} \tag{6.4-2}$$

The orbiter initial weight is given by

$$W_o = W_{BO} - W_{e_B} \tag{6.4-3}$$

where W_{e_B} is determined from Equation (6.1-3).

An iteration is required to determine the orbiter size to complete the mission with the fixed payload. The following equations are solved iteratively until the payload error, ϵ, is within tolerable limits.

$$\mu_o = \text{EXP} \left(\frac{\Delta V_{TOT} - g_r I_{sp} \, \ell n \, \mu_B}{g_r I_{sp_o}} \right) \tag{6.4-4}$$

$$W_{f_o} = \frac{W_o}{\mu_o} \tag{6.4-5}$$

$$W_{p_o} = W_o - W_{f_o} \tag{6.4-6}$$

$$W_{e_o} = f\left(W_{p_o}\right) \text{ or } (6.2\text{-}2) \tag{6.4-7}$$

$$PL = W_{f_o} - W_{e_o} \tag{6.4-8}$$

$$PL_{FIXED} = PL \tag{6.4-9}$$

The following test is made

if $PL < PL_{FIXED}$; $\mu_B = \mu_B - \epsilon$ go to (6.4-1)

$$PL = PL_{FIXED}; \text{ EXIT} \tag{6.4-10}$$

$PL > PL_{FIXED}$; $\mu_B = \mu_B + \epsilon$ go to (6.4-2)

6.5 OPTION 5

This option takes advantage of the equations derived in Section 6.1. The lift-off weight is determined from

$$W_{LO} = \frac{N\left(T_{VAC} - A_e\, p_a\right)}{(T/W)} \tag{6.5-1}$$

and Equations (6.1-1) through (6.1-4) are solved.

6.6 OPTION 6

Option 6 also takes advantage of Equations (6.1-1) through (6.1-4). The trajectory thrust is modified by

$$T_{VAC} = \left(\frac{T}{W}\right)\left(\frac{W_{LO}}{N}\right) + A_e\, p_a \qquad (6.6\text{-}1)$$

6.7 PAYLOAD BOUNDARY CONDITIONS

The payload boundary condition is available in the trajectory module of PADS for use with Phase I sizing problems. This boundary condition is employed for optimal staging problems (rubber stage). The equation for payload boundary condition is the same as Equation (6.4-8). The implications of rubber-stage optimal staging are described in Section 13.

Section 7
INTERFACE WITH SPACE SHUTTLE SYNTHESIS PROGRAM

As was mentioned in Section 1 of this volume, the manned reusable launch vehicle synthesis is performed by the SSSP module in PADS. The synthesis modeling and equations remain essentially the same as documented in Reference 1. There are, however, some minor model changes and programming changes. The model changes will be discussed below. The programming changes, which are mostly related to data communication, are listed herein and discussed in Volume II of this report.

7.1 THRUST SIMULATION CHANGE

The thrust simulation available in the original SSSP program is completely dependent on its trajectory module (GTSM). In PADS, the thrust simulation is likewise related to the TABTOP trajectory module. This thrust simulation is described in Section 3.2 of this report. The information on thrust and fuel flow values must be transferred to the SSSP program in order to calculate fuel weights. This calculation of fuel weights and related quantities is performed in a new subroutine called THRUST.

7.2 ENVIRONMENTAL EFFECTS ON DESIGN

A new thermal protection system weight estimation equation has been added to the SSSP module. It has the form

$$W_{TPS} = \left(Q_L\right)^{r_1} \left(t_L\right)^{r_2} C_{TP_1} + C_{TP_2} \qquad (7.2\text{-}1)$$

where C_{TP_1} and C_{TP_2} are input coefficients and r_1 and r_2 are input exponents. Q_L is the effective heat load (Btu/ft^2) on the entry portion of the trajectory. The effective heat load, Q_L, is computed by considering

the integral of $\dot{Q}$ only when $\dot{Q}$ is above an input threshold value, $\dot{Q}_{Th}$

$$Q_L = \int_{t_1}^{t_2} \dot{Q}\, dt \qquad \dot{Q} > \dot{Q}_{Th} \tag{7.2-2}$$

The quantity t_L in Equation (7.2-1) is

$$t_L = t_2 - t_1 \tag{7.2-3}$$

meaning the duration when $\dot{Q}$ is greater than $\dot{Q}_{Th}$.

A familiar configuration of coefficients for TPS design is

$$\begin{aligned} r_1 &= 1/8 \\ r_2 &= 3/8 \\ C_{TP_1} &= 1. \\ C_{TP_2} &= 0. \end{aligned} \tag{7.2-4}$$

7.3 HUNTING PROCEDURE

A parameter hunting procedure for solving bounded optimization of up to 10 design parameters used in SSSP design synthesis is available in PADS. The technique employed is called Powell's method, originally published in Reference 4. Bounding of free parameters is accomplished through the Box transformation of Reference 5.

Briefly described, the method required no gradients and employs a conjugate direction quadratic ray search to find a minimum. The equations and programming logic for subroutine POWELL are presented in detail in Volume II.

7.4 MISCELLANEOUS CHANGES

Following is a list of key programming changes in SSSP:

A. Input data communication

B. Merging of fly-back range calculations

C. Trajectory program communication

D. Output format

E. Sizing options for solids.

Section 8
AUXILIARY PRINT COMPUTATIONS

This section will detail the auxiliary computations required at each print point. They consist of the instantaneous impact points, inertial Cartesian coordinates, Euler angles, steering angles, and some orbit parameters.

8.1 ADDITIONAL ORBIT PARAMETERS

The majority of the orbit parameters printed at each time point are calculated as described in Section 5. Three parameters printed are not calculated in that section. They are the apogee and perigee velocities and the orbit period. They are determined from the following equations.

$$V_A = \sqrt{GM\left(\frac{2}{R_a} - \frac{1}{a_s}\right)} \qquad (8.1\text{-}1)$$

$$V_P = \frac{R_a}{R_P} V_A \qquad (8.1\text{-}2)$$

$$\tau = 2\pi\sqrt{\frac{a_s^3}{GM}} \qquad (8.1\text{-}3)$$

8.2 INSTANTANEOUS IMPACT POINT

The instantaneous impact point (IIP) is that point on a spherical earth where the vehicle would impact if it continued on its current path. The solution assumes unpowered vacuum flight on a Keplerian orbit.

Certain quantities are calculated and tests are made before the IIP can be calculated in subroutine CRASH. The first test is made to determine if the orbit will intersect the earth. If the perigee radius is greater than the earth

radius, the orbit does not intersect the earth and a message is printed. If the apogee radius is less than the earth radius, an error has been made and a message is printed.

The current true anomaly, ζ, is given by

$$\cos \zeta = \frac{p_r/R - 1}{e} \tag{8.2-1}$$

The true anomaly at the impact point is given by

$$\cos \zeta_{IIP} = \frac{\frac{p_r}{E_R} - 1}{e} \tag{8.2-2}$$

The eccentric anomaly at the current position and the impact point are given by

$$\cos E = \frac{e + \cos \zeta}{1 + e \cos \zeta} \tag{8.2-3}$$

$$\sin E = \sqrt{\frac{\left|i - e^2\right| \sin \zeta}{1 + e \cos \zeta}} \tag{8.2-4}$$

$$E = \tan^{-1}\left(\frac{\sin E}{\cos E}\right) \tag{8.2-5}$$

The impact velocity is given by

$$V_{IIP} = \sqrt{V^2 + \frac{2\,GM}{R}\left(\frac{R - E_R}{E_R}\right)} \tag{8.2-6}$$

The impact elevation angle is given by

$$\cos \gamma_{IIP} = \frac{R\, V_I \cos \gamma_I}{E_R\, V_{IIP}} \tag{8.2-7}$$

The true anomaly to the impact point is

$$\bar{\zeta} = \zeta_{IIP} - \zeta \tag{8.2-8}$$

The geocentric impact latitude is given by

$$\sin \rho_{IIP} = \cos \rho \cos \psi_I \sin \bar{\zeta} + \sin \rho \cos \bar{\zeta} \tag{8.2-9}$$

The azimuth angle at the impact point is given by

$$\sin \psi_{IIP} = \frac{\cos \rho \sin \psi_I}{\cos \rho_{IIP}} \tag{8.2-10}$$

$$\cos \psi_{IIP} = \left(\frac{\cos \bar{\zeta} \sin \rho_{IIP} - \sin \rho}{\sin \bar{\zeta}} \right) \cos \rho_{IIP} \tag{8.2-11}$$

$$\psi_{IIP} = \tan^{-1} \left(\frac{\sin \psi_{IIP}}{\cos \psi_{IIP}} \right) \tag{8.2-12}$$

The longitude increment from the burnout point to the impact point is

$$\sin \bar{\mu} = \frac{\sin \bar{\zeta} \sin \psi_{IIP}}{\cos \rho_{IIP}} \tag{8.2-13}$$

$$\cos \bar{\mu} = \frac{\cos \bar{\zeta} - \sin \rho \sin \rho_{IIP}}{\cos \rho \cos \rho_{IIP}} \tag{8.2-14}$$

$$\bar{\mu} = \tan^{-1} \left(\frac{\sin \bar{\mu}}{\cos \bar{\mu}} \right) \tag{8.2-15}$$

The eccentric anomaly at the impact point is given by

$$\sin E_{IIP} = \sqrt{\frac{\left|1 - e^2\right| \sin \zeta_{IIP}}{1 + e \cos \zeta_{IIP}}} \tag{8.2-16}$$

$$\cos E_{IPP} = \frac{e + \cos \zeta_{IIP}}{1 + e \cos \zeta_{IIP}} \tag{8.2-17}$$

$$E_{IIP} = \tan^{-1}\left(\frac{\sin E_{IIP}}{\cos E_{IIP}}\right) \tag{8.2-18}$$

The time to impact is based on the difference in the time since perigee as determined at the burnout point and at the impact point. These times are given by Kepler's equation as follows:

$$\tau_o = \chi (E - e \sin E) \tag{8.2-19}$$

and

$$\tau_{IIP} = \chi (E_{IIP} - e \sin E_{IIP}) \tag{8.2-20}$$

where

$$\chi = \frac{\tau}{2\pi} = \sqrt{\frac{a_s^3}{GM}} \tag{8.2-21}$$

The time to the impact point from burnout is thus given by

$$\overline{\tau} = \tau_{IIP} - \tau_o \tag{8.2-22}$$

and

$$\overline{\tau} = \tau_{IIP} - \tau_o + \tau \text{ if } \overline{\tau} < o. \tag{8.2-23}$$

The impact longitude is given by

$$\overline{\mu}_{IIP} = \mu_o - \overline{\mu} + \overline{\tau}\,\omega \tag{8.2-24}$$

The impact time is given by

$$T_{IIP} = T_o + \overline{\tau} \tag{8.2-25}$$

The range to the impact point is

$$\overline{S} = E_R\,\overline{\zeta} \tag{8.2-26}$$

The range from the reference latitude and longitude point is given by

$$\cos\,\chi = \cos\rho_R \cos\rho_{IIP} \cos(\mu_R - \mu_{IPP}) + \sin\rho_R \sin\rho_{IIP} \tag{8.2-27}$$

$$S_{IIP} = E_R \cos^{-1}(\cos\chi) \tag{8.2-28}$$

The azimuth angle from the reference point to the impact point is given by

$$A^*_{IIP} = \cos^{-1}\left[\left(\frac{\sin(\mu_R - \mu_{IIP})}{\left|\sin(\mu_R - \mu_{IIP})\right| \sin\left(\frac{S_{IIP}}{E_R}\right)}\right)\left(\cos\rho_R \sin\rho_{IIP} - \cos\rho_{IIP} \sin\rho_R \cos(\mu_R - \mu_{IIP})\right)\right]$$

$$- \frac{\pi}{2}\left[\frac{\sin(\mu_R - \mu_{IIP})}{\left|\sin(\mu_R - \mu_{IIP})\right|} - 1\right] \tag{8.2-29}$$

However, if

$$(\mu_R - \mu_{IIP}) = 0 \text{ and } \begin{cases} \rho_{IIP} < \rho_R: \ A^*_{IIP} = \pi \\ \rho_{IIP} \geq \rho_R: \ A^*_{IIP} = 0 \end{cases} \tag{8.2-30}$$

or, if

$$(\mu_R - \mu_{IIP}) = \pi \text{ and } \begin{cases} \rho_{IIP} < -\rho_R: \ A^*_{IIP} = \pi \\ \rho_{IIP} \geq -\rho_R: \ A^*_{IIP} = 0 \end{cases}$$

These quantities define the attitude and position of the vehicle at the impact point.

8.3 INERTIAL CARTESIAN COORDINATES

The inertial Cartesian coordinates are determined from the inertial longitude, the current latitude, and the radius to the vehicle. The Cartesian frame has the following as its primary planes; the equatorial plane, the plane through the launch site and the north pole, and a plane perpendicular to these planes. The position transformation is given by

$$X_I = R \cos \rho \cos \mu_I \tag{8.3-1}$$

$$Y_I = R \cos \rho \sin \mu_I \tag{8.3-2}$$

$$Z_I = R \sin \rho \tag{8.3-3}$$

The velocity transformation is obtained by differentiation of the above equations to give

$$\dot{X}_I - \dot{R} \cos \rho \cos \mu_I - R \dot{\rho} \sin \rho \cos \mu_I - R\dot{\mu}_I \cos \rho \sin \mu_I \tag{8.3-4}$$

$$\dot{Y}_I = \dot{R} \cos\rho \sin\mu_I - R\dot{\rho} \sin\rho \sin\mu_I + R\dot{\mu}_I \cos\rho \cos\mu_I \tag{8.3-5}$$

$$\dot{Z}_I = \dot{R} \sin\rho + R\dot{\rho} \cos\rho \tag{8.3-6}$$

where

$$\dot{R} = V \sin\gamma \tag{8.3-7}$$

$$\dot{\rho} = \frac{V}{R} \cos\gamma \cos\psi \tag{8.3-8}$$

$$\dot{\mu} = \frac{V}{R} \frac{\cos\gamma \sin\psi}{\cos\rho} \tag{8.3-9}$$

8.4 RELATIVE EULER ANGLES

The relative Euler angles of the vehicle axes are obtained from the vehicle state by establishing a Cartesian coordinate system at the vehicle center of gravity and performing the following rotations in order:

1. Rotate about the z axis through the azimuth angle ψ.
2. Rotate about the y axis through the flight path angle γ.
3. Rotate about x through the bank angle ϕ.
4. Rotate about y through the angle-of-attack, α.

The matrix product of these rotations results in the matrix of direction cosines between the vehicle axes and the relative Cartesian set.

$$D = \begin{pmatrix} \text{Direction} \\ \text{Cosines} \end{pmatrix}_R = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8.4-1}$$

Given the direction cosines, the Euler angles are calculated in subroutine DCTOE for both yaw-pitch-roll and pitch-yaw-roll sequences of rotation. The equations for these angles are YAW, PITCH, ROLL

$$\Theta_{PITCH} = \tan^{-1}\left(\frac{D(1,3)}{\sqrt{1 - D(1,3)^2}}\right) \tag{8.4-2}$$

$$\Theta_{YAW} = \tan^{-1}\left(\frac{D\ (1,\ 2)}{D\ (1,\ 1)}\right) \tag{8.4-3}$$

$$\Theta_{ROLL} = \tan^{-1}\left(\frac{D\ (2,\ 3)}{D\ (3,\ 3)}\right) \tag{8.4-4}$$

PITCH, YAW, ROLL

$$\Theta_{PITCH} = \tan^{-1}\left(\frac{-D\ (1,\ 3)}{D\ (1,\ 1)}\right) \tag{8.4-5}$$

$$\Theta_{YAW} = \tan^{-1}\left(\frac{-D\ (1,\ 2)}{\sqrt{1 - D\ (1,\ 2)^2}}\right) \tag{8.4-6}$$

$$\Theta_{ROLL} = \tan^{-1}\left(\frac{-D\ (3,\ 2)}{D\ (2,\ 2)}\right) \tag{8.4-7}$$

where D (I, J) is a location in the direction cosine matrix given in Equation (8.4-1).

8.5 INERTIAL EULER ANGLES

The inertial Euler angles are obtained by a right multiplication of the relative direction cosine matrix [(Equation (8.4-1)] by the transformation

$$D' = \begin{pmatrix} \cos\mu & \sin\mu & 0 \\ -\sin\mu & \cos\mu & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8.5-1}$$

to obtain the matrix of inertial direction cosines

$$\begin{pmatrix} \text{direction} \\ \text{cosines} \end{pmatrix}_I = \begin{pmatrix} \text{direction} \\ \text{cosines} \end{pmatrix}_R (D') \tag{8.5-2}$$

The same equations used to determine the relative Euler angles (subroutine DCTOE) are used to determine the inertial Euler angles from Equation (8.5-2).

8.6 STEERING ATTITUDE ANGLES

Two steering attitude angles are calculated for output. The first angle is the elevation angle from the local horizontal to the vehicle centerline in the plane determined by the local vertical and the velocity vector. The second angle is the azimuth angle measured from north to the vehicle centerline projection on the local horizontal plane. The equations for these angles are:

$$\Theta_P = \gamma + \sin^{-1} (\sin \alpha \cos \phi) \tag{8.6-1}$$

$$\Theta_\ell = \psi + \sin^{-1} (\sin \alpha \sin \phi) \tag{8.6-2}$$

Section 9
CONTROL VECTOR SOLUTION – NON-OPTIMAL CONTROL

To integrate the equations of motion it is necessary to know the control vector

$$U = (T, \delta_E, \alpha, \phi)^T \qquad (9.0\text{-}1)$$

Because we are concerned with endo- as well as exo-atmospheric trajectories, this can be a formidable problem, even if we overlook the added complication of determining the optimal steering angles, α and ϕ. (See Sections 11, 12, 16, and 17 for discussion of this problem.)

For example, suppose we want the vehicle to perform a "gravity turn" while it balances its aerodynamic moment and, at the same time, conforms to a total acceleration limit (not an unlikely circumstance). Under these conditions, none of the in-plane control quantities, T, δ_E, or α, can be explicitly evaluated; each is dependent upon the other two. In other words, we are confronted with a situation that requires the solution of a system of nonlinear simultaneous algebraic equations. Were this the only example in which this predicament arose, it would be of little burden, but, unfortunately, it is only one of many.

The difficulty of the problem is compounded by the fact that the governing conditions may change at any time. For example, the thrust may have to be instantaneously throttled in order to satisfy a total acceleration limit, or the angle of attack may have to switch from one mode of flight to another.

Hence, to determine the vector U, the above difficulties compel us to adopt the rather oblique and formalistic approach described below. In doing so, we reap at least one important supplementary benefit – later program modifications will be greatly facilitated by the formalism we introduce now.

9.1 NON-OPTIMAL BANK ANGLE

Presently, there are only two non-optimal conditions for the bank angle, ϕ. One of them is that $\phi = 0$. The other is the vertical rise/pitchover mode. In both cases, an explicit equation for ϕ can be obtained. In the former case it is simply

$$\phi = 0 \tag{9.1-1}$$

and, in the latter, it is

$$\phi = \tan^{-1}(k_\psi / k_\gamma) \tag{9.1-2}$$

where

$$k_\psi = V \cos\gamma \sin\rho \left(\frac{V}{R \cos\rho} \cos\gamma \sin\psi + 2\omega \right)$$

$$+ \omega \cos\rho \, (R\omega \sin\rho \sin\psi - 2V \cos\psi \sin\gamma) \tag{9.1-3}$$

and

$$k_\gamma = \left(\frac{V^2}{R} - g \right) \cos\gamma - V\dot{\gamma}^* +$$

$$+ \omega \cos\rho \left[2V \sin\psi + R\omega (\cos\rho \cos\gamma + \sin\rho \cos\psi \sin\gamma) \right] \tag{9.1-4}$$

As a result, we can exclude this component of the U vector from the formalistic treatment and restrict our attention to the problem of determining the in-plane control vector

$$w = (T, \delta_E, \alpha)^T \tag{9.1-5}$$

9.2 IN-PLANE CONTROL DETERMINATION WHEN α IS NON-OPTIMAL

Suppose the vehicle is at some arbitrary point of a non-optimal portion of its flight. Let

$$K = (K_1, K_2, K_3)^T \tag{9.2-1}$$

be the system of governing equations for the choice of w at this particular point. In general, K will be a function of the state and the in-plane control; i.e.,

$$K = K(y, w) \tag{9.2-2}$$

The in-plane control vector (value of w) will be determined by solving:

$$K(y, w) = 0 \tag{9.2-3}$$

To do so, we make use of the well-known Newton-Raphson iteration, which runs as follows:

Starting with some initial guess for w, e.g., w_o, compute an increment Δw from the equation

$$\Delta w = -K_w^{-1}(w_o)\, K(w_o) \tag{9.2-4}$$

(ALGCON, BLGCON)

where $K_w(w_o)$ is the matrix of explicit partials of K with respect to w evaluated at $w = w_o$; i.e.

$$K_w(w_o)_{ij} = K_{i_{w_j}}(w_o), \qquad i = 1, 2, 3, ; \quad j = 1, 2, 3 \tag{9.2-5}$$

If

$$\left\| \Delta w \right\| = \sum_{i=1}^{3} \left| \Delta w_i \Big/ w_i \right| \leq \epsilon \qquad \text{(see Note)} \qquad (9.2\text{-}6)$$

(ALGCON, BLGCON)

we are done; otherwise, increment w_o by Δw and repeat the process until Equation (9.2-6) is satisfied.

A rigorous discussion of the necessary conditions for the convergence of this iteration would be inappropriate here. Suffice it to say that its success is largely dependent upon the guess w_o and the behavior of K_w.

Presently, PADS' policy in choosing w_o is the following:

At the first point of the trajectory,

$$w_o(t_o) = (10^6,\ 0,\ 0)^T \qquad (9.2\text{-}7)$$

at all corner points,

$$w_o(t^+) = w(t^-) \qquad (9.2\text{-}8)$$

and at the interior points of a subarc

$$w_o(t) = w\,(t - \Delta t) \qquad (9.2\text{-}9)$$

where Δt is the integration step size and w is the converged value of the in-plane control vector. So far, this policy has been adequate, but the need for a more elaborate one can always arise.

Note: $\epsilon = 10^{-12}$ CDC 6500 version

$\epsilon = 10^{-7}$ Univac 1108 version

9.3 PARTIAL DERIVATIVES OF THE NON-OPTIMAL CONTROL VECTOR

Both the steepest descent and QL modules of PADS require the evaluation of the total first partials of the in-plane control quantities at each point of the trajectory. Steepest descent needs them on its backward integration in order to construct the adjoint coefficients (see Section 12). The QL module needs them to evaluate the Euler-Lagrange equations (see Section 16).

For non-optimal portions of the flight, it follows from the implicit function theorem that these partials are given by

$$D_y \cdot w = -K_w^{-1} K_y \tag{9.3-1}$$

where D_y is the total partial derivative operator

$$D_y = \left(\frac{\partial}{\partial y_1}, \frac{\partial}{\partial y_2} \cdots\cdots \frac{\partial}{\partial y_n} \right) \tag{9.3-2}$$

K_w is the matrix of explicit partial derivatives of K with respect to w, and K_y is the matrix of explicit partial derivatives of K with respect to y. Of course, both K_w and K_y are evaluated at the converged value of w.

Because the QL module of PADS employs the method of quasi-linearization to solve the multipoint boundary value problem that arises from the calculus of variations, it also requires the total second partial derivatives of w with respect to y. Differentiating Equation (9.3-1) with respect to an arbitrary state variable y_i yields

$$D_{y_i} \cdot (Dy \cdot w) = -K_w^{-1} \left[Kyy_i + Y_i + \left(K_{wy_i} + X_i \right) D_y \cdot w \right] \qquad i = 1, \ldots n \tag{9.3-3}$$

where the j-th column of the matrix Y_i is given by

$$Y_i^{(j)} = K_{wy_i} D_{y_i} \cdot w, \quad j = 1, \ldots\ldots, n \tag{9.3-4}$$

and the k-th column of the matrix X_i is given by

$$X_i^{(k)} = K_{ww_k} Dy_i \cdot w, \quad k = 1, 2, 3 \tag{9.3-5}$$

Section 10
CONTROL LAWS

In Section 2.4, the concept of the control vector U and its subset, the in-plane control vector,

$$w^T = (T, \delta_E, \alpha) \tag{10.0-1}$$

was introduced. From prior descriptions of the simulations in PADS and discussions in Section 9, it is apparent that w can be calculated explicitly only in special circumstances. In this section, the equations corresponding to K_3 in the algebraic constraint vector introduced in Section 9 will be presented. The first sections describe the non-optimal control capabilities in PADS. The later sections describe the calculations necessary for in-flight bounded control and state function constraints.

10.1 VERTICAL RISE AND PITCHOVER

During the vertical rise or pitchover control mode, the bank angle and angle of attack are fully determined. The equation for bank angle is given in Equation (9.1-2). The equation for K_3 involves k_γ defined in Equation (9.1-4) is,

$$K_3 = (T \sin(\alpha - \delta_E) + L - D_b \sin\alpha) \cos\phi + m k_\gamma \tag{10.1-1}$$

This equation plus associated partial derivatives may be found in subroutine BL4 (steepest descent) and AL4 (Q.L.).

10.2 CONSTANT ANGLE OF ATTACK

The constant angle-of-attack control mode is used for a number of different situations in PADS. This simplest case is when the zero angle-of-attack

control mode is chosen. The form of constraint function, K_3, in all such cases is,

$$K_3 = \alpha - C_\alpha = 0 \tag{10.2-1}$$

The zero α control mode has

$$C_\alpha = 0 \tag{10.2-2}$$

The most significant use of this control mode is in the steepest descent control computation. As shown in Equation (12.1-23), the control correction $\delta\alpha$ is computed using the steepest descent equations. This increment is added to the nominal control

$$\alpha = \alpha_{OLD} + \delta\alpha \tag{10.2-3}$$

This new value α is inserted into Equation (5.2-1) by letting

$$C_\alpha = \alpha \tag{10.2-4}$$

This control constraint is likewise used under any circumstance when the maximum angle-of-attack limit is reached. Suppose α^* is the calculated angle of attack either in the steepest descent or quasi-linearization program and

$$\left|\alpha^*\right| > \alpha_{max} \quad (\alpha_{max} > 0) \tag{10.2-5}$$

Then

$$C_\alpha = \alpha_{max} (\text{sign } (\alpha^*)) \tag{10.2-6}$$

10.3 GRAVITY TURN CONTROL

The gravity turn control establishes a balance of aerodynamic and propulsive forces so that there is no net force normal to the flight path. Referring to Equations (2.3-2) and (2.3-3), this will occur when

$$T \sin (\alpha - \delta_E) + L - D_b \sin \alpha = 0, \text{ or } a^X = 0 \qquad (10.3\text{-}1)$$

The form of the constraint K_3 is the same.

$$K_3 = T \sin (\alpha - \delta_E) + L - D_b \sin \alpha = 0, \text{ or } K_3 = a^X = 0 \qquad (10.3\text{-}2)$$

10.4 MAXIMUM LIFT CONTROL BOUNDARY

The maximum lift force magnitude may be instantaneously constrained by employing this control mode. From Equation (3.1-6), the uncorrected lift is

$$L_u = \left(C_{L_O} + C_{L_\alpha}\, \alpha\right) q\, \delta_{Ref} \qquad \text{for aerodynamic option 1 or 3} \qquad (10.4\text{-}1)$$

and

$$L_u = C_L\, q\, \delta_{REF} \qquad \text{for aerodynamic option 2.} \qquad (10.4\text{-}2)$$

The desired maximum lift

$$C_{u_L} = L_{max} \text{ sign } (L_u) \qquad (10.4\text{-}3)$$

Hence, the form of the K_3 function is

$$K_3 = L_u - C_{u_L} = 0 \qquad (10.4\text{-}4)$$
(AL3, BL3)

F

10.5 UNPOWERED TOTAL ACCELERATION LIMIT

When the unpowered total acceleration limit is applied, the angle of attack may be solved for, and the bank angle may still be optimized. The equation for K_3 is:

$$K_3 = L^2 + D^2 - (g_{max}\, W)^2 = 0 \text{ or } K_3 = (a^x)^2 + (a^V)^2 - (g_{max}\, W)^2 \qquad (10.5\text{-}1)$$

This equation and it associated partials are programmed in subroutines AL5 and BL5.

10.6 STATE INEQUALITY CONTROL MODES

When state instantaneous inequality functions are imposed on the trajectory, first of all, a corner point must be included at which the bounding function reaches its maximum value. After reaching its maximum value, the vehicle will fly the boundary until the optimized control will tend to tangentially fly the vehicle off the boundary. While the vehicle is on the boundary it is actually flying, a non-optimal control law which must be included in the K_3 set.

There are three state inequality bounding control laws in PADS. These are listed below with their corresponding inequality limit.

Control Law	Corresponding Inequality
$\dot{q} = 0$	Maximum dynamic pressure limit
$\ddot{Q} = 0$	Maximum heating rate
$\dot{R}_{ey} = 0$	Maximum Reynolds number

For the dynamic pressure rate, $\dot{q}$, control law, the equation K_3 is

$$K_3 = V \dot{R} \frac{\partial \rho_a}{\partial h} + 2\rho_a \dot{V} = 0 \qquad (10.6\text{-}1)$$

(AL7, BL7)

For the heating rate limit, $\ddot{Q}$, control law, the equation is

$$K_3 = \frac{(V^{1.15})\,\dot{R}\,\frac{\partial \rho_a}{\partial h}}{6.3} + \rho_a \dot{V} = 0 \qquad (10.6\text{-}2)$$
(AL8, BL8)

For the Reynolds number rate, $\dot{R}_{ey}$, control law, the governing equation is

$$K_3 = \left(\frac{\frac{\partial \rho_a}{\nu \partial h} - \frac{\partial \nu}{\partial h}\rho_a}{\nu^2} \right) \dot{R} + \frac{\rho_a}{\nu} \dot{V} = 0 \qquad (10.6\text{-}3)$$
(AL9, BL9)

where ν is the viscosity.

Section 11
POSING THE PROBLEM FOR STEEPEST DESCENT

The steepest descent method as applied to solving for the control vector U will be described in the following steps.

A. Derive the influence functions for steering angle, arc time duration, and initial state effects on arbitrary end boundary conditions.

B. Using these influence functions, develop the equations for the closed-loop steepest descent algorithm.

C. Show how the closed-loop steepest descent algorithm generates steering angle and trajectory parameter corrections to first force the resulting trajectory to satisfy problem constraints and second drive a payoff quantity to its approximate optimal value.

D. Show how the steepest descent solution may be transformed to approximate the exact Lagrange multipliers for use in starting the quasi-linearization module of PADS.

11.1 ADJOINTS AND INFLUENCE FUNCTIONS

The equation

$$\dot{y} = f_i \{y, U_i, t_i\} \quad 0 \le t_i \le \tau_i \tag{11.1-1}$$

defines the dynamical system during arc i subject to an arc cut-off function

$$\Omega_i \{ y, w_i, t_i \} = 0 \tag{11.1-2}$$

which determines τ_i. The control vector, U, contains the optimizing steering vector u (see Note).

$$U_i = (T, \delta_E, u_i)^T \tag{11.1-3}$$

Note: The solution for U on arcs with suboptimal control (K(U, y) = 0) is explained in Sections 9 and 10.

Then, the optimization problem resolves into determining a set of U histories (one per arc) and ω_i adjustable parameters that will optimize some function of state Φ at the terminus of an arc while constraining additional functions, Ψ, at the termini of that or other arcs.

If the trajectory during arc i as defined in Equation (11.1-1) is perturbed, the perturbed solution can be related to the nominal result at any time t

$$\dot{y}' = \bar{f}_i + \delta\dot{y} \tag{11.1-4}$$

Discarding higher-order terms, this may be expressed in operator notation as a Taylor series which, after subtracting the nominal value, becomes

$$\delta\dot{y}' = D_y^* \cdot \bar{f}_i \, \delta y + \frac{\partial \bar{f}_i}{\partial u_i} \delta u_i \tag{11.1-5}$$

where $D_y^* \cdot \bar{f}_i$, and $\partial\bar{f}/\partial u_i$ are matrices of partial derivatives (See Note).

The end conditions for the i^{th} stage are given by the equation

$$\Psi_i = \Psi\left\{ y \,\middle|\, t_i = \tau_i \right\} \tag{11.1-6}$$

Equation (11.1-5) gives the perturbation effect of the state and control on the derivative $\dot{y}'$; however, we really need to know the effect of state and control changes on the end condition Ψ_i.

Note: D_y^* is a partial derivative operator which assumes the steering control vector elements are constant.

We postulate the desired form of this relation as

$$\delta\Psi_i = \left(\lambda^{\Psi_i}\right)^T \delta y \Big|_t + E\{\delta u_i\} \tag{11.1-7}$$

where the matrix of influence functions λ^{Ψ_i} relates the effect of changes in the state at time t on $\delta\Psi_i$. The function E relates the effect of control changes up to time t on $\delta\Psi_i$. For the sake of this discussion, assume final time is fixed and under this stipulation take time derivatives of Equation (11.1-7), resulting in

$$\dot{\delta\Psi}_i = 0 = \left(\lambda^{\Psi_i}\right)^T \dot{\delta y}\Big|_t + \left(\dot{\lambda}^{\Psi_i}\right)^T \delta y + \dot{E} \tag{11.1-8}$$

Now, substitute Equation (11.1-5) into (11.1-8)

$$\left(\lambda^{\Psi_i}\right)^T \left(D_y^* \cdot \bar{f}_i\, \delta y + \frac{\partial \bar{f}_i}{\partial u_i}\, \delta u_i\right) + \left(\dot{\lambda}^{\Psi_i}\right)^T \delta y + \dot{E} = 0 \tag{11.1-9}$$

or

$$\left[\left(\lambda^{\Psi_i}\right)^T D_y^* \cdot \bar{f}_i + \left(\lambda^{\Psi_i}\right)^T\right] \delta y + \left(\lambda^{\Psi_i}\right)^T \frac{\partial \bar{f}_i}{\partial u_i}\, \delta u_i + \dot{E} = 0 \tag{11.1-10}$$

This equation must hold for any δy and therefore the coefficient of δy must equal zero

$$\left(\lambda^{\Psi_i}\right)^T D^* \cdot \bar{f}_i + \left(\dot{\lambda}^{\Psi_i}\right)^T = 0 \tag{11.1-11}$$

or in more familiar form, take the transpose to get the adjoint differential equations (see Note)

$$\dot{\lambda}^{\Psi_i} = -\left[D_y^* \cdot \bar{f}_i\right]^T \lambda^{\Psi_i} \qquad (11\text{-}1\text{-}12)$$

Equation (11.1-12) is used to solve for the influence coefficients subject to the boundary condition

$$\lambda^{\Psi_i}\Big|_{t_i = \tau_i} = \frac{\partial \Psi_i}{\partial y}\Big|_{t_i = \tau_i} \qquad (\tau_i \text{ fixed}) \qquad (11.1\text{-}13)$$

The last two terms in Equation (11.1-10) may be used to solve for the control influence function

$$\dot{E} = \left(-\lambda^{\Psi_i}\right)^T \frac{\partial \bar{f}_i}{\partial u_i}\, \delta u_i \qquad (11.1\text{-}14)$$

If the equation is integrated backwards from τ_i to t, then

$$E\Big|_t - E\Big|_{\tau_i} = -\int_{\tau_i}^{t} \left(\lambda^{\Psi_i}\right)^T \frac{\partial \bar{f}_i}{\partial u_i}\, \delta u_i dt \qquad (11.1\text{-}15)$$

There can be no influence of control changes at τ_i and therefore $E\big|_{\tau_i} = 0$

Note: On arcs where control is subject to the algebraic constraint, Equation (11.1-12) becomes

$$\dot{\lambda}^{\Psi_i} = -\left(D_y^* \cdot f_i - D_u \cdot f_i\, D_y \cdot K\,(D_u \cdot K)^{-1}\right)^T \lambda^{\Psi_i}$$

Using the following definition for the impulse response function

$$\Lambda^{\Psi_i} = \left(-\lambda^{\Psi_i}\right)^T \frac{\partial \bar{f}_i}{\partial u_i} \tag{11.1-16}$$

Equation (11.1-15) becomes

$$E\Big|_t = \int_{\tau_i}^{t} \Lambda^{\Psi_i}\, \delta u_i\, dt \tag{11.1-17}$$

If the final arc time, τ_i, is allowed to vary, the perturbation in the end condition may be described by combining results in Equation (11.1-7)

$$\delta\Psi_i = \left(\lambda^{\Psi_i}\right)^T \delta y\Big|_{t=o} + \int_{\tau_i}^{o} \Lambda^{\Psi_i}\, \delta u_i\, dt + \dot{\Psi}_i\, d\tau_i \tag{11.1-18}$$

Equation (11.1-18) is a vector equation which contains an unknown final time variation, $d\tau_i$. However, the scalar cutoff function Ω_i defined in Equation (11.1-2) should determine the final time variation. Since Equation (11.-1-18) is a general form, we may describe the variation of the scalar cutoff function in a like manner.

$$\delta\Omega_i = \left(\lambda^{\Omega_i}\right)^T \delta y\Big|_{t_o} + \int_{\tau_i}^{o} \Lambda^{\Omega_i}\, \delta u_i dt + \dot{\Omega} d\tau_i \tag{11.1-19}$$

Since $\Omega_i = 0$ and $\delta\Omega_i = \delta\omega_i$, $d\tau_i$ may be solved for as follows:

$$d\tau_i = \frac{1}{\dot{\Omega}_i}\left[\delta\omega_i - \left(\lambda^{\Omega_i}\right)^T \delta y\Big|_{t=o} - \int_{\tau_i}^{o} \Lambda^{\Omega_i}\, \delta u_i dt\right] \tag{11.1-20}$$

Defining

$$\lambda^{\Psi\Omega_i} = \lambda^{\Psi_i} - \frac{\dot{\Psi}_i}{\dot{\Omega}_i} \lambda^{\Omega_i} \qquad (11.1\text{-}21)$$

and substituting Equation (11.1-20) into Equation (11.1-18) yields the relation

$$\delta\Psi_i = \left(\lambda^{\Psi\Omega_i}\right)^T \delta y\Big|_{t=o} + \int_{\tau_i}^{o} \Lambda^{\Psi\Omega_i} \delta u dt + \frac{\dot{\Psi}_i}{\dot{\Omega}_i} \delta\omega i \qquad (11.1\text{-}22)$$

The term $\lambda^{\Psi\Omega_i}$ satisfies the adjoint differential equations if its boundary condition is

$$\lambda^{\Psi\Omega_i}\Big|_{\tau_i} = \frac{\partial\Psi_i}{\partial y}\Big|_{\tau_i} - \frac{\dot{\Psi}_i}{\dot{\Omega}_i} \frac{\partial\Omega_i}{\partial y}\Big|_{\tau_i} \qquad (11.1\text{-}23)$$

This equation is programmed in subroutine ADIC operating in conjunction with subroutine PDBC which supplies the matrix $\partial\Psi_i/\partial y\big|_{\tau_i}$ for nonlinear functions.

11.2 MULTI-ARC PROBLEMS

Section 11.1 has developed the influence functions for a single arc problem. In this section the formulation is extended to multi-arc problems with only terminal constraints (Reference 6).

Thus, in an i stage problem

$$d\Psi = \delta\Psi_i\Big|_{\tau_i} \qquad (11.2\text{-}1)$$

means that all constraint perturbations are determined at the end of the i^{th} arc. Now examine the effect of control and parameter changes in the last two arcs of the trajectory on the constraint perturbation.

$$d\Psi = \left(\lambda^{\Psi_i\Omega_{i-1}}\right)^T \delta y \Big|_o + \int_{\tau_{i-1}}^{o} \Lambda^{\Psi_i\Omega_{i-1}} \delta u_{i-1}\, dt + \int_{\tau_i}^{o} \Lambda^{\Psi_i\Omega_i} \delta u_i\, dt$$

$$+ \left(\lambda^{\Psi_i\Omega_{i-1}}\right)^T \dot{y} \frac{\delta\omega_{i-1}}{\dot{\Omega}_{i-1}}\Bigg|_{\tau_{i-1}^-} + \frac{\dot{\Psi}_i}{\dot{\Omega}_i}\delta\omega_i\Bigg|_{\tau_i} \qquad (11.2\text{-}2)$$

Following the form of Equation (11.1-23), the boundary condition on the adjoints at the next to last arc corner point is:

$$\lambda^{\Psi_i\Omega_{i-1}}\Big|_{\tau_{i-1}^-} = \lambda^{\Psi_i\Omega_i}\Big|_{\tau_{i-1}^+} - \frac{\left(\lambda^{\Psi_i\Omega_i}\right)^T \dot{y} \frac{\partial\Omega_{i-1}}{\partial y}}{\dot{\Omega}_{i-1}}\Bigg|_{\tau_{i-1}^-}, \qquad \begin{matrix}(11.2\text{-}3)\\(\text{ADID3A})\end{matrix}$$

noting that

$$\dot{\Psi}_i\Big|_{\tau_{i-1}^-} = \left(\lambda^{\Psi_i\Omega_i}\right)^T \dot{y}\Big|_{\tau_{i-1}^-} \qquad (11.2\text{-}4)$$

for correspondence with (11.1-23).

Equation (11.2-2) may be put in completely general terms by summing over the index j

$$d\Psi = \sum_{j=1}^{i} \int_{\tau_j}^{o} \Lambda^{\Psi_i\Omega_j} \delta u_j dt + \sum_{j=1}^{i-1} \left(\lambda^{\Psi_i\Omega_{j+1}}\right)^T \dot{y} \frac{\delta\omega_j}{\dot{\Omega}_j}\Bigg|_{\tau_j^-}$$

$$+ \frac{\dot{\Psi}_i}{\dot{\Omega}_i}\Bigg|_{\tau_i} \delta\omega_i + \left(\lambda^{\Psi_i\Omega_i}\right)^T \delta y\Big|_0 \qquad (11.2\text{-}5)$$

which is the general form for control and parameter sensitivities with boundary conditions defined only at the trajectory terminus.

Equation (11.2-3) is programmed in subroutine ADID3A. Arc time sensitivities are programmed in subroutine STAU.

11.3 MIXED BOUNDARY CONDITIONS

A more general class of problems arise when constraints are not completely defined at the trajectory terminus; that is, $d\Psi \neq \delta\Psi_i\Big|_{\tau_i}$.

The simplest example of this is the case where the boundary condition is an explicit function of arc times

$$\Psi_i = \Psi_i\{\tau_j\} \tag{11.3-1}$$

We refer back to Equation (11.2-4), which now becomes

$$\dot{\Psi}_i\Big|_{\tau_{i-1}^-} = \left(\lambda^{\Psi_i\Omega_i}\right)^T \dot{y}\Big|_{\tau_{i-1}^-} + \frac{\partial\Psi}{\partial\tau_{i-1}} \tag{11.3-2}$$

The equation for the particular adjoint discontinuity corresponding to Equation (11.2-3) is

$$\lambda^{\Psi_i\Omega_{i-1}}\Big|_{\tau_{i-1}^-} = \lambda^{\Psi_i\Omega_i}\Big|_{\tau_{i-1}^+} - \frac{\left[\left(\lambda^{\Psi_i\Omega_i}\right)^T \dot{y} + \frac{\partial\Psi_i}{\partial\tau_{i-1}}\right]\frac{\partial\Omega_{i-1}}{\partial y}}{\dot{\Omega}_{i-1}}\Bigg|_{\tau_{i-1}^-} \tag{11.3-3}$$

For the case where

$$d\Psi_i = k \sum_{j=1}^{i} \delta\tau_j \,, \tag{11.3-4}$$

$$\frac{\partial \Psi_i}{\partial \tau_{i-1}} = k \tag{11.3-5}$$

This corresponds to three different boundary conditions in PADS:

A. Elapsed time $t_e = \sum_{j=1}^{i} \tau_i$

B. Inertial longitude $\mu_I = \mu + \omega t_e$

C. Longitude of ascending node (see Section 5).

Equation (11.3-3) is programmed in ADID3A.

11.4 INTERMEDIATE BOUNDARY CONDITIONS

Suppose some components of the $d\Psi$ vector are completely defined at an arc corner point – say, arc s – prior to the i^{th} arc. This implies that the Ψ boundary condition vector can be partitioned

$$\Psi = (\Psi_s \quad \Psi_i)^T \tag{11.4-1}$$

For the partitioned terms, Equation (11.2-5) becomes

$$d\Psi_s = \sum_{j=1}^{s} \int_{\tau_j}^{o} \Lambda^{\Psi_s \Omega_j} \delta u j dt + \sum_{j=1}^{s-1} \left(\lambda^{\Psi_s \Omega_{j+1}} \right)^T \cdot y \left. \frac{\delta \omega_j}{\dot{\Omega}_j} \right|_{\tau_j^-}$$

$$+ \left. \dot{\Psi}_s / \dot{\Omega}_s \right|_{\tau_s^-} \delta\omega_s + \lambda^{\Psi_s \Omega_1} \delta y \Big|_o \tag{11.4-2}$$

Likewise, the boundary conditions on the intermediate constraint adjoints are

$$\lambda^{\Psi_s \Omega_s}\Big|_{\tau_s} = \frac{\partial \Psi_s}{\partial y}\Big|_{\tau_s} - \frac{\dot{\Psi}_s}{\dot{\Omega}_s}\,\frac{\partial \Omega_s}{\partial y}\Big|_{\tau_s} \qquad \begin{matrix}(11.4\text{-}3)\\(\text{ADICB3})\end{matrix}$$

whereas the same arc corner the remaining adjoints are adjusted as in Equation (11.2-3) or (11.3-2).

$$\lambda^{\Psi_i \Omega_s}\Big|_{\tau_s^-} = \lambda^{\Psi_i \Omega_{s+1}}\Big|_{\tau_s^+} - \frac{\left(\lambda^{\Psi_i \Omega_{s+1}}\right)^T \dot{y}\,\dfrac{\partial \Omega_s}{\partial y}}{\dot{\Omega}_s}\Bigg|_{\tau_s^-} \qquad \begin{matrix}(11.4\text{-}4)\\(\text{ADICB3})\end{matrix}$$

These equations are programmed in subroutine ADICB3.

11.5 BRANCHED TRAJECTORY BOUNDARY CONDITIONS

Figure 11-1 illustrates the arrangement of a typical branched trajectory.

CR155-I

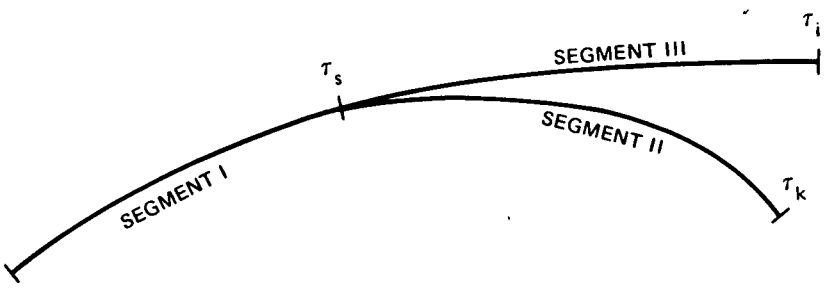


Figure 11-1. Branched Trajectory

τ_s defines the branch point and vector Ψ is partitioned into Ψ_i and Ψ_k. Clearly all parameter and control changes during segment I influence both Ψ_i and Ψ_k. However perturbations during segment II can have no effect on Ψ_i and vice versa.

The adjoint boundary conditions are:

At τ_i (segment III)

$$\lambda^{\Psi_i \Omega_i}\Big|_{\tau_i} = \frac{\partial^{\Psi_i}}{\partial y}\Big|_{\tau_i} - \frac{\dot{\Psi}_i}{\dot{\Omega}_i}\,\frac{\partial^{\Omega_i}}{\partial y}\Big|_{\tau_i} \qquad \begin{matrix}(11.5\text{-}1)\\(\text{ADICB3})\end{matrix}$$

and at τ_k (segment II)

$$\lambda^{\Psi_k \Omega_k}\Big|_{\tau_k} = \frac{\partial^{\Psi_k}}{\partial y}\Big|_{\tau_k} - \frac{\dot{\Psi}_k}{\Omega_k}\,\frac{\partial^{\Omega_k}}{\partial y}\Big|_{\tau_k} \qquad \begin{matrix}(11.5\text{-}2)\\(\text{ADIC})\end{matrix}$$

Equation (11.5-1) is solved and the adjoints then associated with Ψ_i are integrated backwards from τ_i to τ_s^+. This integration is stopped and then Equation (12.5-2) is solved and the adjoints associated with Ψ_k are integrated back to τ_s^+ at which point the Ψ_i and Ψ_k vectors and adjoint matrix are merged. Adjoint discontinuities (if appropriate) are calculated using Equation (11.3-3) or (11.2-3) and the integration may proceed to initial time.

The equations for the branched adjoint boundary conditions are contained in subroutine ADICB3.

Section 12

CLOSED-LOOP STEEPEST DESCENT ALGORITHM

In the prior section, the influence functions for control, initial conditions and staging parameters have been derived. In this section these influence functions will be incorporated into the closed-loop, steepest descent algorithm formulation.

12.1 USE OF INFLUENCE FUNCTIONS

We repeat Equation (11.2-5) making the following substitutions:

$$S_{\tau_k}^{\Psi_i} = \left. \frac{\left(\lambda^{\Psi_i \Omega_k} \right)^T \dot{y}}{\dot{\Omega}_k} \right|_{\tau_k^-} \qquad (12.1\text{-}1)$$

Note that $\dot{\Omega} \equiv 1$ for a timed arc cut-off; hence Equation (12.1-1) becomes

$$S_{\tau_k}^{\Psi_i} = \left. \left(\lambda^{\Psi_i \Omega_k} \right)^T \dot{y} \right|_{\tau_k^-} \qquad (12.1\text{-}2)$$
(STAU)

Therefore, using the new notation

$$d\Psi^i = \left(\lambda^{\Psi_i \Omega_i} \right)^T \delta y \Big|_o + \sum_{j=1}^{i} \int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} \delta u_j dt + \sum_{j=1}^{i} S_{\tau_j}^{\Psi_i} d\tau_j \qquad (12.1\text{-}3)$$

We keep in mind that the $d\Psi_i$ vector may be partitioned for branching or intermediate constraints and now also stipulate that the last element in this vector be the performance index, $d\Phi$, as shown below:

$$d\Psi = \left(d\Psi_s \quad d\Psi_i \quad d\Phi \right)^T \tag{12-1-4}$$

We further simplify the notation of Equation (12. 1-3) by combining free initial conditions and staging time sensitivity terms

$$S^{\Psi i} = \left(S_y^{\Psi_i} \quad S_{\tau_j}^{\Phi_i} \right)^T \tag{12. 1-5}$$

$$\delta p = \left(\delta y \Big|_{t_o} \quad \delta \tau_j \right)^T \tag{12. 1-6}$$

where

$$S_y^{\Psi i} = \lambda_y^{\Psi \Omega_1} \Big|_{t_o} \quad \text{is a scalar relation.} \tag{12. 1-7 (SINIT)}$$

Equation (12. 1-3)then becomes

$$d\Psi_i = \left(\lambda^{\Psi_i \Omega_1} \right)^T \delta y \Big|_o + \sum_{j=1}^{i} \int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} \delta u_j dt + S^{\Psi_i} \delta p \tag{12. 1-8}$$

It is apparent that for a given finite order $d\Psi_i$ vector there are an infinite number of solutions to Equation (12. 1-8); moreover, if particular sensitivities are small, the associated contributions of parameter or control perturbations may be negligible relative to more sensitive contributors. To account for the discrepancies in sensitivities we assign a weight to each of the control and parameter sensitivities.

Let W be a diagonal control sensitivity weighting matrix. For this program, it has two elements:

$$[W] = \begin{pmatrix} 1 & 0 \\ 0 & W_\phi \end{pmatrix} \tag{12.1-9}$$

Let Y be a diagonal parameter sensitivity weighting matrix.

$$[Y] = \begin{pmatrix} Y_{P_1} & 0 & 0 & 0 \\ 0 & Y_{P_2} & 0 & 0 \\ 0 & 0 & \cdot & 0 \\ 0 & 0 & & \cdot \end{pmatrix} \tag{12.1-10}$$

To account for the multiplicity of solutions to Equation (12.1-8), we will seek the one that forces perturbations along the direction of steepest descent to satisfy all the constraints. To do this, the control and parameter perturbations must be maximized.

The metric of control and parameter perturbations given below is to be maximized according to the development of Reference 7.

$$(dP)^2 = \sum_{j=1}^{i} \left(\int_{\tau_j}^{o} \delta u_j^T \, W \, \delta u j dt \right) - \delta p^T Y \delta p \tag{12.1-11}$$

To the $(dP)^2$ metric we adjoin the problem constraints through the Lagrange multipliers μ^T. Using variational calculus, we will maximize the resulting functional given below.

$$J = (dP)^2 + \mu^T \left\{ d\Psi_i - \left(\lambda^{\Psi_i \Omega_i} \right)^T \delta y \Big|_{t_o} - \sum_{j=1}^{i} \left[\int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} \delta u_j dt \right] \right.$$

$$\left. - S^{\Psi_i} \delta p \right\} \qquad (12.1\text{-}12)$$

Substituting Equation (12.1-11) into Equation (12.1.12) and combining terms gives

$$J = \sum_{j=i}^{i} \left[\int_{\tau_j}^{o} \left(\delta u_j^T W \delta u_j - \mu^T \Lambda^{\Psi_i \Omega_j} \delta u_j \right) dt \right]$$

$$+ \mu^T \left[d\Psi_i - \left(\lambda^{\Psi_i \Omega_i} \right)^T \delta y \Big|_{t_o} \right] - \delta p^T Y \delta p - \mu^T S^{\Psi_i} \delta p$$

(12.1-13)

In order for J to be maximized (stationary), δJ must equal zero. We therefore take the first variation. Note that W, Y and μ are constants.

$$\delta J = 0 = \sum_{j=1}^{i} \left[\int_{\tau_i}^{o} \left(\delta u_j^T W \delta(\delta u_j) + \delta\left(\delta u_j^T\right) W \delta u_j - \mu^T \Lambda^{\Psi_i \Omega_j} \delta(\delta uj) \right) dt \right]$$

$$- \delta p^T Y \, \delta(\delta p) - \delta(\delta p)^T Y \, \delta p - \mu^T S^{\Psi_i} \delta(\delta p) \qquad (12.1\text{-}14)$$

The terms $\delta(\delta u^T) W \delta u$ and $\delta(\delta p^T) Y \delta p$ are equal to their respective transposes since both W and Y are diagonal matrices.

$$\left(\delta(\delta u^T)W\delta u\right)^T = \delta u^T W^T \delta(\delta u)$$

But $W^T = W$ and therefore

$$\left(\delta(\delta u^T)W\delta u\right)^T = \delta u^T W \delta(\delta u)$$

Hence

$$\left(\delta(\delta u^T)W\delta u\right)^T = \delta u^T W \delta(\delta u) \qquad (12.1\text{-}15)$$

and likewise

$$\delta(\delta p^T)Y\delta p = \delta p^T Y \delta(\delta p) \qquad (12.1\text{-}16)$$

Substituting Equations (12.1-15) and 12.1-16 into Equation (12.1-14) yields

$$\delta J = \sum_{j=1}^{i} \left[\int_{\tau_j}^{o} \left(2\delta u_j^T W \delta(\delta u_j) - \mu^T \Lambda^{\Psi_i \Omega_j} \xi(\delta u_j) \right) dt \right]$$

$$- \left(2\delta p^T Y + \mu^T S^{\Psi_i} \right) \zeta(\zeta p) = 0 \qquad (12.1\text{-}17)$$

The coefficients of $\delta(\delta u)$; and $\delta(\delta p)$ must equal zero for arbitrary perturbations. Thus

$$\delta u_j^T = 1/2\mu^T \Lambda^{\Psi_i \Omega_j} W^{-1} \text{ or } \delta\mu_j = 1/2 W^{-1} \left(\Lambda^{\Psi_i \Omega_j} \right)^T \mu \qquad (12.1\text{-}18)$$

$$\delta p^T = 1/2\mu^T S^{\Psi_i} Y^{-1} \text{ or } \delta p = 1/2 Y^{-1} \left(S^{\Psi_i} \right)^T \mu \qquad (12.1\text{-}19)$$

The substitution of Equations (12. 1-18) and (12. 1-19) into Equation (12. 1-8) will allow the solution for μ

$$d\Psi_i = \left(\lambda^{\Psi_i \Omega_l} \right)^T \delta y \Big|_{t_o} + \left[\sum_{j-1}^{i} \left(1/2 \int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} W^{-1} \left(\Lambda^{\Psi_i \Omega_i} \right)^T dt \right) \mu \right]$$

$$- 1/2 S^{\Psi_i} Y^{-1} \left(S^{\Psi_i} \right)^T \mu \qquad (12.1\text{-}20)$$

or

$$\mu = 2 \left[\sum_{j=1}^{i} \left(\int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} W^{-1} \left(\Lambda^{\Psi_i \Omega_j} \right)^T dt \right) - S^{\Psi_i} Y^{-1} \, S^{\Psi_i \, T} \right]^{-1}$$

$$x \left(d\Psi_i - \left(\lambda^{\Psi_i \Omega_l} \right)^T \delta y \Big|_{t_o} \right) \qquad (12.1\text{-}21)$$

or, if the bracketed term is called the A matrix,

$$\mu = 2 \, A^{-1} \left(d\Psi_i - \left(\lambda^{\Psi_i \Omega_l} \right)^T \delta y \Big|_{t_o} \right) \qquad (12.1\text{-}22)$$

Substituting Equation (12. 1-22) into Equations (12. 1-18) and (12. 1-19) gives

$$\delta u_j = W^{-1} \left(\Lambda^{\Psi_i \Omega_j} \right)^T A^{-1} \left(d\Psi_i - \left(\lambda^{\Psi_j \Omega_j} \right)^T \delta y \Big|_{t_o} \right) \qquad (12.1\text{-}23) \text{ (MTXI)}$$

and

$$\delta p = - Y^{-1}\left(S^{\Psi_i}\right)^T A^{-1}\left(d\Psi_i - \left(\lambda^{\Psi_i\Omega_j}\right)^T \delta y \Big|_{to}\right) \qquad \begin{matrix}12.1\text{-}24)\\(\text{MTXI})\end{matrix}$$

If the A matrix is inverted only at the initial point, in Equation (12.1-23), the time history of impluse response functions will yield a history of control corrections which will under normal circumstances drive the $d\Psi$ vector to zero. However, it is advantageous to consider that as the new trajectory is integrated, the difference in state between the nominal and new trajectory at each arc time point does affect the end constraints, $d\Psi_i$. It is from this consideration that the closed-loop steepest descent algorithm arises (Reference 7). In this algorithm, the time history of the control contribution to A is computed, and the parameter sensitivity contributions are included up to the time where the parameter is adjusted. An example of this is given below where the control correction δu, at arc time t during arc k is shown

$$\delta u_k\{t\} = W^{-1}\left(\Lambda^{\Psi_i\Omega_k}\{t\}\right)^T \left(A\{t\}\right)^{-1}$$

$$\left(d\Psi_i - \left(\lambda^{\Psi_i\Omega_k}\{t\}\right)^T \delta y\{t\}\right) \qquad (12.1\text{-}25)$$

where

$$\delta y\{t\} = y(t) - \bar{y}(t) \qquad (12.1\text{-}26)$$

This equation and the corresponding one for parameter corrections are computed in subroutine MTX3A.

As described in Section 11.4 and 11.5, the $d\Psi$ vector may be partitioned for intermediate constraints or branching. The A matrix is symmetric and of the same order as the number of problem boundary conditions. If during a

portion of the trajectory only part of the $d\Psi$ vector applies, the rows and columns of A associated with the active constraints are compressed into a new matrix, called B, before inversion. This is shown by the following:

Assume there are n problem constraints. There are m on the first branch and m+1 through n on the second branch. We need examine only the diagonal elements of A (because of symmetry). Let vector D' = diag (A) then, D corresponds element wise to the vector $d\Psi$

$$D' = \begin{bmatrix} A_{11} \\ A_{22} \\ \cdot \\ \cdot \\ \cdot \\ A_{mm} \\ A_{m+1,\,m+1} \\ \cdot \\ \cdot \\ \cdot \\ A_{nn} \end{bmatrix} \longleftrightarrow \begin{bmatrix} d\Psi_1 \\ d\Psi_2 \\ \cdot \\ \cdot \\ \cdot \\ d\Psi_m \\ d\Psi_{m+1} \\ \cdot \\ \cdot \\ \cdot \\ d\Psi_n \end{bmatrix} \qquad (12.1\text{-}27)$$

On the first branch of the trajectory the B matrix is defined by

$$B = \begin{bmatrix} A_{11} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & A_{mm} \end{bmatrix} \qquad (12.1\text{-}28)$$

and on the second branch,

$$B = \begin{bmatrix} A_{m+1,\,m+1} & & \\ & \ddots & \\ & & A_{n,\,n} \end{bmatrix} \qquad (12.1\text{-}29)$$

Clearly, on the main trunk of the trajectory (see Figure 11-1)

$$B \equiv A \qquad (12.1\text{-}30)$$

The matrix compression logic corresponding to the above description is programmed in subroutine MTX3A.

12.2 PAYOFF IMPROVEMENT

The concept of the expandable constraint vector is used to good advantage in PADS. During the first few iterations of a solution, only the constraint elements are contained in the $d\Psi$ vector. As soon as these elements are driven as close to zero as desired, the $d\Psi$ vector is expanded to contain the first improvement in payoff quantity, $d\Phi$. Assuming this payoff or some fraction of it can be achieved on the next trial trajectory, a method is needed for estimating the payoff improvement for subsequent trials.

As a result of asking for a $d\Phi$, a history of δu and δp changes result which will define the control metric $(dP)^2$ given in Equation (12.1-11). Substituting Equations (12.1-23) and (12.1-24) into Equation (12.1-11) and using the definition of matrix A,

$$(dP)^2 = \left(d\Psi_i\right)^T A^{-1} d\Psi_i \qquad (12.2\text{-}1)$$

It is now convenient to break A and Ψ_i into minors. Let n be the number of constraints

$$(d\Psi_i) = \left(d\Psi_i' d\Phi\right)^T \tag{12.2-2}$$

$$\left[A\right]^{-1} = \begin{bmatrix} & & A_{1,n+1} \\ & M & \vdots \\ & & A_{n,n+1} \\ A_{n+1,n} & \cdots\cdots & A_{n+1,n+1} \end{bmatrix} \tag{12.2-3}$$

and

$$N^T = (A_{n+1,n} \cdots\cdots A_{n+1,n+1}) \tag{12.2-4}$$

We now rewrite Equation (12.2-1).

$$(dP)^2 = (d\Psi_i' d\phi)\, A^{-1} \begin{pmatrix} d\Psi_i' \\ d\Phi \end{pmatrix} \tag{12.2-5}$$

We can now expand Equation (12.2-5) in terms of the partitioned matrix and vectors.

$$dP^2 = A_{n+1,n+1}\, d\Phi^2 + 2N^T d\Psi_i' d\Phi + \left(d\Psi_i'\right)^T M d\Psi_i' \tag{12.2-6}$$

This is a scalar quadratic equation in $d\Phi$ which has the solutions.

$$d\Phi = \frac{-N^T d\Psi_i'}{A_{n+1,n+1}} \pm \frac{\sqrt{(dP)^2 - (d\Psi_i')^T \left(M - \frac{NN^T}{An+1,n+1}\right) d\Psi_i'}}{A_{n+1,n+1}} \tag{12.2-7}$$

It should be noted that the initial value of the metric, dP^2, is calculated by the approximation

$$(dP)^2 = d\Phi^2 A_{n+1,\,n+1} \qquad (12.2\text{-}8)$$

This approximation assumes that the remaining elements of the $d\Psi_i$ vector are all near zero. The equations above are programmed in subroutine PAYØ2.

12.3 ADJOINT EQUATIONS

The adjoint differential equations are programmed in subroutine ADEQ3A. They appear in engineering notation in the same order as computed; that is, an ordered adjoint vector is one row in the adjoint variable matrix. Each row corresponds to one element in the (n+1) order $d\Psi$ vector. Hence, for the r^{th} constraint ($1 \le r \le n+1$), the adjoint variable or influence function for velocity, V, during a particular arc s is defined for convenience as

$$\lambda^*_v = \lambda_v^{\Psi_r \Omega_s} \qquad (12.3\text{-}1)$$

Subscript partial derivative notation will be used here. For example,

$$\dot{\gamma}_v = D_v \cdot \dot{\gamma} = \frac{\partial \dot{\gamma}}{\partial v} \qquad (12.3\text{-}2)$$

It should be noted that the adjoint differential equations include only terms that can be non-zero.

$$\dot{\lambda}^*_v = -\left(\dot{V}_v \lambda^*_v + \dot{\gamma}_v \lambda^*_\gamma + \dot{h}_v \lambda^*_h + \dot{m}_v \lambda^*_m + \dot{\psi}_v \lambda^*_\psi + \dot{\rho}_v \lambda^*_\rho + \dot{\mu}_v \lambda^*_\mu + \dot{Q}_v \lambda^*_Q\right) \qquad (12.3\text{-}3)$$

$$\dot{\lambda}^*_\gamma = -\left(\dot{V}_\gamma \lambda^*_v + \dot{\gamma}_\gamma \lambda^*_\gamma + \dot{h}_\gamma \lambda^*_h + \dot{\psi}_\gamma \lambda^*_\psi + \dot{\rho}_\gamma \lambda^*_\rho + \dot{\mu}_\gamma \lambda^*_\mu\right) \qquad (12.3\text{-}4)$$

$$\dot{\lambda}_h^* = -\left(\dot{V}_h \lambda_v^* + \dot{\gamma}_h \lambda_\gamma^* + \dot{m}_h \lambda_m^* + \dot{\psi}_h \lambda^* + \dot{\rho}_h \lambda_\rho^* + \dot{\mu}_h \lambda_\mu^* + \dot{Q}_h \lambda_Q^*\right) \tag{12.3-5}$$

$$\dot{\lambda}_m^* = -\left(\dot{V}_m \lambda_v^* + \dot{\gamma}_m \lambda_\gamma^* + \dot{m}_m \lambda_m^* + \dot{\psi}_m \lambda_\psi^*\right) \tag{12.3-6}$$

$$\dot{\lambda}_\psi^* = -\left(\dot{V}_\psi \lambda^*_v + \dot{\gamma}_\psi \lambda_\gamma^* + \dot{\psi}_\psi \lambda_\psi^* + \dot{\rho}_\psi \lambda_\rho^* + \dot{\mu}_\psi \lambda_\mu^*\right) \tag{12.3-7}$$

$$\dot{\lambda}_\rho^* = -\left(\dot{V}_\rho \lambda_v^* + \dot{\gamma}_\rho \lambda_\gamma^* + \dot{\psi}_\rho \lambda_\psi^* + \dot{\mu}_\rho \lambda_\mu^*\right) \tag{12.3-8}$$

$$\dot{\lambda}_\mu^* = 0 \tag{12.3-9}$$

$$\dot{\lambda}_Q^* = 0 \tag{12.3-10}$$

The partial derivatives of the equations of motion with respect to the states are not presented in detail here. They are programmed in subroutine PDY3A. The correspondence between engineering notation and FORTRAN symbol follows rules which facilitate interpretation. These are shown in Table 12.3-1.

Table 12.3-1

EXAMPLES OF FORTRAN NOTATION
FOR PARTIAL DERIVATIVES

Engineering Symbol	FORTRAN Symbol	Engineering Symbol	FORTRAN Symbol
$\dot{V}_v$	VDV	$\dot{\rho}_v$	ØDV
$\dot{V}_\gamma$	VDG	$\dot{\mu}_h$	UDR
$\dot{V}_h$	VDR	$\dot{Q}_v$	HTDV
$\dot{V}_m$	VDM	$\dot{Q}_h$	HTDR
$\dot{V}_\psi$	VDP	$\dot{V}_\alpha$	VDA
$\dot{V}_\rho$	VDØ	$\dot{\psi}_\alpha$	PDA
$\dot{V}_\mu$	VDU	$\dot{\gamma}_\phi$	GDPH
$\dot{\gamma}_v$	GDV	$\dot{\psi}_\phi$	PDPH
$\dot{h}_v$	HDV	a_v^v	AVV $\left(\frac{\partial a^v}{\partial v}\right.$, see Section 2$\left.\right)$
$\dot{m}_m$	MDM	a_h^γ	AGR $\left(\frac{\partial a^\gamma}{\partial h}\right.$, see Section 2$\left.\right)$
$\dot{\psi}_\rho$	PDØ		

Section 13
STAGING OPTIMIZATION

The rubber-stage design optimization equations which may be employed for Phase I sizing problems are described in this section. This type of optimization problem has a performance index (payload), which is a function of state values at more than one arc corner point, and as such is classed as a mixed boundary-condition problem.

Equation (6.4-8) indicates that the payload is simply the difference between the final orbiter weight (burnout) and the empty structure weight.

$$PL = W_{f_o} - W_{e_o} \tag{13-1}$$

It is clear that W_{f_o} is really the weight of the vehicle at final time

$$m\Big|_{\tau_i} = W_{f_o} \Big/ g_r \quad . \tag{13-2}$$

The orbiter structural weight is given in Equation (6.2-2) as a function of orbiter propellant.

$$W_{e_o} = f_o\left(W_{p_o}\right) \tag{13-3}$$

We need to express the payload, PL, in terms of the mass (state vector element) at corner points where the variation of mass is non-zero and an explicit dependence of payload on mass exists.

The weight of propellant used during the orbiter burn is

$$W_{p_o} = W_o - W_{f_o} \tag{13-4}$$

At booster burnout, the booster empty weight is dropped, resulting in

$$W_o = W_{BO} - W_{eB} \tag{13-5}$$

where W_{BO} is the booster burnout weight and W_{eB} is the booster empty weight defined by

$$W_{eB} = f_B\left(W_{p_B}\right) \tag{13-6}$$

W_{p_B} is the booster propellant weight, which in turn is related to the difference between the vehicle gross lift-off weight and booster burnout weight.

$$W_{p_B} = W_{L.O.} - W_{BO} \tag{13-7}$$

Now, substitute Equation (13-7) into (13-6):

$$W_{eB} = f_B(W_{L.O.} - W_{BO}) \tag{13-8}$$

Then, substitute Equation (13-8) into (13-5):

$$W_o = W_{BO} - f_B(W_{L.O.} - W_{BO}) \tag{13-9}$$

(13-9) into (13-4):

$$W_{p_o} = W_{BO} - f_B(W_{L.O.} - W_{BO}) - W_{f_o} \tag{13-10}$$

(13-10) into (13-3):

$$W_{eo} = f_o\left[W_{BO} - f_B(W_{L.O.} - W_{BO}) - W_{f_o}\right] \tag{13-11}$$

and, finally, (13-11) into (13-1):

$$PL = W_{f_o} - f_o\left[W_{BO} - f_b \left(W_{L.O.} - W_{BO}\right) - W_{f_o} \right] \qquad \begin{matrix}(13\text{-}12)\\(PAYLOD)\end{matrix}$$

This is what is needed to relate payload to the weight at the end of orbiter burn and at the end of boost.

In order to construct the adjoint initial conditions or transversality, conditions at the injection point, the explicit partials of payload with respect to mass are required.

At orbiter injection

$$\frac{\partial PL}{\partial m}_{\tau_i} = (1 + f_o') \, g_r \qquad \begin{matrix}(13\text{-}13)\\(PDBC)\\(PDBCQL)\end{matrix}$$

where

$$f_o' = \frac{\partial f_o}{\partial W_{p_o}}$$

At booster burnout

$$\frac{\partial P_L}{\partial m}_{\tau_{BO}} = - f_o' \, (1 + f_B') \, g_r \qquad \begin{matrix}(13\text{-}14)\\(ADJUMP)\end{matrix}$$

where

$$f_B' = \frac{\partial f_B}{\partial W_{p_o}}$$

Note that $W_{L.O.}$ is invariant and f_o, f_o', f_B and f_B' are evaluated at nominal values of W_{p_o} and W_{p_B}, respectively.

Section 14

LAGRANGE MULTIPLIER TRANSFORMATION

This analysis shows how the matrix of adjoint variables can be transformed into an approximation for the Euler-Lagrange multipliers. These transformation equations are actually a useful by-product of proving that in the limit, the steepest descent solution satisfies the necessary conditions of the calculus of variations for the Mayer problem.

We need to rephrase the problem of steepest descent in a manner similar to that presented by Denham and Bryson in Reference 8. We subdivide the constraint vector and its corresponding sensitivities into $d\Psi_i$ and $d\phi$ and use as a goal the optimization of $d\Phi$. We need to adjoin the expression for dP^2 and the constraints. The resulting expression is [(see Equation (12.1-12)].

$$
\begin{aligned}
d\Phi = &\left(S^{\Phi}\right)^T \delta p + \sum_{j-1}^{i} \int_{\tau_j}^{o} \Lambda^{\Phi\Omega_j} \delta u_j dt \\
&+ \nu^T \left[d\Psi_i - \left(S^{\Psi i}\right)^T \delta p - \sum_{j=1}^{i} \int_{\tau_j}^{o} \Lambda^{\Psi_i \Omega_j} \delta u_j dt \right] \\
&+ \mu \left[(dP)^2 - \delta p^T Y \delta p - \sum_{j=1}^{i} \int_{\tau_j}^{o} (\delta u_j)^T W \delta u_j dt \right]
\end{aligned}
\tag{14-1}
$$

If this expression is regrouped

$$d\Phi = \left[(S^{\Phi})^T - \nu^T (S^{\Psi i})^T - \mu (\delta p)^T Y \right] \delta p$$

$$+ \sum_{j=1}^{i} \int_{\tau_j}^{o} \left(\Lambda^{\Phi \Omega_j} - \nu^T \Lambda^{\Psi_i \Omega_j} - \mu (\delta u_j)^T W \right) \delta u_j dt$$

$$+ \nu^T d^{\Psi}_i + \mu (dP)^2 \qquad (14\text{-}2)$$

We take the first variation of $d\Phi$

$$\delta(d\Phi) = \left[\left(S^{\Phi} \right)^T - \nu^T \left(S^{\Psi i} \right)^T - 2\mu (\delta p)^T Y \right] \delta(\delta p)$$

$$+ \sum_{j=1}^{i} \int_{\tau_j}^{o} \left(\Lambda^{\Phi \Omega_j} - \nu^T \Lambda^{\Psi_i \Omega_j} - 2\mu (\delta u_j)^T W \right) \delta(\delta u_j)\, dt$$

$$+ \nu\, \delta(d\Psi_i) + \mu\, \delta(dP)^2 \qquad (14\text{-}3)$$

$(dP)^2$ is a prespecified quantity; therefore, its variation will be zero. Likewise, $d\Psi_i$ is considered fixed and its variation is zero. For arbitrary variation of δu_j and δp, Equation (14-3) can only equal zero if

$$\left(S^{\Phi} \right)^T - \nu^T \left(S^{\Psi i} \right)^T - 2\mu (\delta u_j)^T Y = 0 \qquad (14\text{-}4)$$

and

$$\Lambda^{\Phi\Omega_j} - \nu^T \Lambda^{\Psi_i\Omega_j} - 2\mu(\delta u_j)^T W = 0 \tag{14-5}$$

The parameter and control corrections are then

$$(\delta p)^T = \left[(S^{\Phi})^T - \nu^T s^{\Psi i}\right]\frac{Y^{-1}}{2\mu} \tag{14-6}$$

$$(\delta u)^T = \left[\Lambda^{\Phi\Omega_j} - \nu^T \Lambda^{\Psi_i\Omega_j}\right]\frac{W^{-1}}{2\mu} \tag{14-7}$$

We may rewrite Equation (14-7) as (see Equation 11.1-16)

$$(\delta u_j)^T = \left[-\lambda^{\Phi\Omega_j^T} + \nu^T \left(\lambda^{\Psi_i\Omega_j}\right)^T\right]\frac{D_u \cdot \overline{f}\, W^{-1}}{2\mu} \tag{14-8}$$

From the calculus of variations, we know the necessary condition that

$$-\lambda^T D_u \cdot f = 0 \tag{14-9}$$

Here, λ is the Euler-Lagrange multiplier on the exact extremal. On the solution, δu_j is likewise identically zero. We may then infer that the linear transformation

$$\lambda^T \approx \left(\lambda^{\Phi\Omega_j}\right)^T - \nu^T\left(\lambda^{\Psi_i\Omega_j}\right)^T \tag{14-10}$$

is a good approximation of the Euler-Lagrange multiplier.

By substituting Equations (14-6) and (14-7) into Equation (14-2) we may calculate ν^T. In terms of the expressions used in Section 12.2 [see Equation (12.2-3)].

$$\nu^T = [N]^T [M] \tag{14-11}$$

Hence,

$$\lambda^T = \left(\lambda^{\Phi\Omega_j}\right)^T - [N]^T [M] \left(\lambda^{\Psi i\Omega_j}\right)^T \tag{14-12}$$

where the adjoints are all functions of time and N and M are evaluated only at the beginning of the trajectory (at the end of the adjoint solution). This computation is performed in subroutine TRAN3. The stored adjoints are transformed and stored on sequential file for use later in the quasi-linearization portion of the program.

Section 15
STEEPEST DESCENT NUMERICAL ANALYSIS TECHNIQUES

This section of the report discusses some of the more important numerical analyses and convergence techniques employed in the steepest descent portion of PADS. Other aspects of the programming techniques used are found in Volume II.

15.1 NUMERICAL INTEGRATION, STORAGE RETRIEVAL, AND STEERING COMPUTATION

The numerical integration method used in the steepest descent algorithm is standard fourth-order Runge Kutta, as described in Reference 9. The same method is used for both forward trajectory and adjoint differential equation integration. RKTA3A and RKTB3A are the integration subroutines.

The basic integration cycle includes evaluation of the derivative twice at the mid-compute interval and twice at the end of the compute interval. During the integration of the adjoint differential equations, it is essential that data for the state vector be accurate at each evaluation of the adjoint derivatives. This is necessary because the adjoint derivatives are functions of the state on the nominal, $\overline{f}$, trajectory. To accommodate this requirement, the adjoint integration has specially adjusted integration intervals to match up exactly with the time integration intervals of the nominal trajectory. An additional consideration arises at the mid-compute interval since the state is only an estimate there. To get a better representation of the state at the mid-compute interval, the following equation is used (ENTRY CORVAR in Subroutine REU3).

$$y_{1/2} = y_o + \left(\frac{f\left\{y_o\right\} + f\left\{y_o + f(y_o)\frac{\Delta t}{2}\right\}}{4} \right) \Delta t \qquad (15.1\text{-}1)$$

where $y_{1/2}$ is the state vector at the beginning of an integration interval, $f\{y_o\}$ is the derivative at that point, and $f\{y_o + f(y_o)\frac{\Delta t}{2}\}$ is the derivative evaluated at an estimated mid-point state.

During the forward trajectory integration, the following data are stored at each mid- and end-integration interval in preparation for the adjoint integration: (1) y, the state vector; (2) α, ϕ, the steering vector; (3) i, the arc number; (4) i_p, the phase number; (5) K_1 flag, the option flag that tells which governing equation is to be used for thrust; (6) K_3 flag, the option flag that tells which governing equation is to be used for α; and (7) arc, phase, and elapsed time.

During the integration of the adjoint differential equations, the forward trajectory data are retrieved at each mid- and full-interval to permit the calculation of the necessary partial derivatives. As the adjoint integration proceeds, a number of quantities are stored in preparation for computing the control and parameter corrections on the next trial trajectory. These include the adjoint matrix, $\lambda^{\Psi_i \Omega_j}$; the impulse response matrix, $\Lambda^{\Psi_i \Omega_j}$; and the A matrix excluding parameter contributions (see Equations (12. 1-21) and (12. 1-22)).

At the mid-compute interval of the adjoint integration, no refinements of the adjoint variables are made. However, the impulse response matrix is refined before storing it. The reason for this selection will be given in the next few paragraphs.

The optimized steering computation on the trial trajectory using retrieved data from the adjoint solution is a very critical aspect of the program. The technique that is used was developed in the course of trying a great number of different approaches and has proved to be the most stable and efficient.

At the beginning of a compute interval at arc time t, the following information is available.

A. y(t), the current state vector (trial trajectory)

B. $y_{old}(t)$, the old state vector (nominal trajectory)

C. $\lambda^{\Psi_i \Omega_j}(t)$, the adjoint matrix

D. u_{old}, p_{old}, the old steering and parameter values (nominal trajectory

E. $S^{\Psi}I$, parameter sensitivities for parameters that have not yet been perturbed

F. $\Lambda^{\Psi_i \Omega_j}(t)$, the impulse response matrix.

Equation (12.1-23) is to be evaluated. The first calculation performed is

$$\delta y(t) = y(t) - y_{old}(t) \qquad \text{(15.1-2) (MTX)}$$

Then, if any parameters are still to be perturbed

$$[SS] = S^{\Psi_i}{}_Y{}^{-1} \left(S^{\Psi_i}\right)^T \qquad \text{(15.1-3) (MTX)}$$

At this point, the A and $[SS]$ matrices are simultaneously added and compressed according to the extent of the $d\Psi_i$ vector

$$B(t) = A(t) - [SS] \qquad \text{(15.1-4) (MTX)}$$

Then B(t) is inverted using a symmetric matrix inversion method (Reference 10).

Using the compression logic again, the augmented constraint vector is calculated.

$$\vec{RR}(t) = d\Psi_i - \left(\lambda^{\Psi_i \Omega_j}(t)\right)^T \delta y(t) \qquad \text{(15.1-5) (MTX)}$$

The first part of the calculation provides the $\vec{TR}$ vector.

$$\vec{TR}(t) = B(t)^{-1} \vec{RR}(t) \qquad \text{(15.1-6) (MTX)}$$

At arc time t, the control may be calculated as

$$u(t) = u_{old}(t) + W^{-1} \Lambda^{\Psi_i \Omega_j}(t)\, \vec{TR}(t) \qquad (15.1\text{-}7)$$
(MTXI)

It is necessary to calculate the control at the mid-interval and end-interval.

The vector $\vec{TR}(t)$ is not recalculated. The equations are therefore

$$u(t + \frac{\Delta t}{2}) = u_{old}(t + \frac{\Delta t}{2}) + W^{-1} \Lambda^{\Psi_i \Omega_j}(t + \frac{\Delta t}{2})\, TR(t)$$

and

$$u(t + \Delta t) = u_{old}(t + \Delta t) + W^{-1} \Lambda^{\Psi_i \Omega_j}(t + \Delta t)\, TR(t) \qquad (15.1\text{-}8)$$

As mentioned in Section 12.1, the vector $\vec{TR}$ is recalculated only if the so-called closed-loop control mode is operating. Once the integration marches past the input elapsed time where the feedback stops. the recalculation of $\vec{TR}$ simultaneously stops and $\vec{TR}$ remains constant for the remainder of the trajectory.

Optimized arc time corrections are calculated up until the arc time is actually changed. At this point, the sensitivity is dropped from the S^{Ψ} matrix and its contribution to $[SS]$ is eliminated.

15.2 ARC CUT-OFF TECHNIQUE

It is essential that the arc j cut-off function, Ω_j, be as numerically close to zero as possible and likewise that the state vector at τ_j be precise.

When

$$\Omega_j = t - \tau_j \qquad (15.2\text{-}1)$$

obviously the satisfaction of $\Omega_j = 0$ is trivial; however, when

$$\Omega_j = G_j(y(t)) - \Omega_j^* \qquad (15.2\text{-}2)$$

where Ω^* is the desired cut-off and G is an arbitrary nonlinear function of the relative states, some special considerations are necessary.

At time t during arc j, one can estimate the value of G at the next integration interval

$$G_j(t + \Delta t) = G_j(t) + \dot{G}_j(t)\,\Delta t \tag{15.2-3}$$

If

$$G_j(t + \Delta t) \geq \Omega^*_j, \tag{15.2-4}$$

then the cut-off condition is detected. It remains to get a good estimate of the integration interval required to satisfy

$$G_j(t + \Delta t') = \Omega^*_j \tag{15.2-5}$$

Let

$$H = G_j(t) - G_j(t - \Delta t) \tag{15.2-6}$$

and

$$H' = \Omega^*_j - G_j(t) \tag{15.2-7}$$

We may get a second order approximation of $\Delta t'$. Let

$$A' = 1 + \frac{H'}{H} \tag{15.2-8}$$

$$A2 = \left(\frac{H'}{H}\right)^2 \tag{15.2-9}$$

$$C = A2(A') \tag{15.2-10}$$

$$B = A'(1. - A2) \tag{15.2-11}$$

and

$$A = 1. - B \tag{15.2-12}$$

Then

$$\Delta t' = A\,(t - \Delta t) + B(t) + \frac{CH}{\dot{G}_j\{t\}} - t \tag{15.2-13}$$

This equation is programmed in subroutine DTF3.

The integration will then march out an interval $\Delta t'$ at which time it will be further refined.

Let

$$K = G_j\{t + \Delta t'\} - G(t) \tag{15.2-14}$$

$$K' = \Omega^*_j - G_j\{t + \Delta t'\} \tag{15.2-15}$$

$$R' = \frac{K'}{K} \tag{15.2-16}$$

$$B' = (R')^2\,(3 - 2R') \tag{15.2-17}$$

$$C' = 1 - B' \tag{15.2-18}$$

$$D' = R'\,(R' - 1)^2 \tag{15.2-19}$$

and

$$E' = (R')^2\,(R' - 1) \tag{15.2-20}$$

Then, the refined time estimate for τ is

$$\tau = C't + B'(t + \Delta t') + K \frac{D'}{\dot{G}_j\{t\}} + \frac{E'}{\dot{G}_j\{t + \Delta t'\}} \qquad (15.2\text{-}21)$$

The entire state vector is refined through a similar third-order interpolation formula (Hermitian interpolation). This is programmed in subroutine YREF3.

Experience with this algorithm has been very good with the exception of one type of situation. Suppose inequality [see Equation (15.2-4)] does not detect the cut-off. This would mean that at the next compute interval, the integration would have to march backwards. This is not permitted owing to limitations of storage retrieval logic and therefore the cut-off equation will not be satisfied. The only remedy for this occurrence is to reduce the integration interval size to minimize the likelihood that it will happen.

The fact that previous time-point data are used for this algorithm makes it imperative that more than two integration intervals be used in each arc.

15.3 SOLUTION CONVERGENCE LOGIC

The basic sequence of convergence in the steepest descent program is

A. Integrate first nominal trajectory
B. Satisfy problem constraints
C. Improve payoff only after problem constraints are satisfied
D. Continue payoff improvement until predicted improvement is very small or iterations exceed maximum number.

15.3.1 First Nominal Trajectory

The first nominal trajectory uses an input control history (on cards) or a prior stored solution-control history (see Volume III). This trajectory is integrated and its state and control are stored as described in Section 15.1. Upon completing the terminal arc cut-off, the nominal constraint misses are calculated and the constraint vector $d\psi_i$ is established.

15.3.2 Satisfaction of Constraints

One or more iterations are normally required to satisfy the problem constraints. After the first nominal trajectory is completed (iteration 1) the first adjoint solution is integrated and stored. Using this stored data, the program will attempt to drive all of the constraint misses $d\Psi_i$ to zero. If these misses are relatively small to start with, they may be driven down to a smaller size or even less than tolerance levels. If, on the other hand, they are large, the control computation may diverge or the constraint misses may get even larger. In the event that divergence occurs, the asked-for constraint corrections will all be halved and another trajectory, using the same stored data, will be integrated. The prime criterion for a satisfactory constraint pass is

$$|d\Psi_i| \leq \text{Tol} \qquad (15.3\text{-}1)\ (\text{TEST})$$

where Tol is a vector of input constraint tolerances or

$$|d\Psi_i| \leq |d\Psi_i|_{\text{old}} \qquad (15.3.2)\ (\text{TEST})$$

This means that all elements of $|d\Psi_i|$ have decreased since the last nominal trajectory. If some have increased and others decreased, the final test is:

$$\sum_{i=1}^{n} \left(\frac{d\Psi_i}{\text{tol}}\right)^2 < \sum_{i=1}^{n} \left(\frac{d\Psi_i}{\text{tol}}\right)^2_{\text{old}} \qquad (15.3\text{-}3)\ (\text{TEST})$$

where n = number of constraints.

This means that the sum square metric of relative constraint misses has decreased since the last trajectory.

If after five halving trials these tests are not satisfied, the program will stop and go on to the next case.

If, on the other hand, the test in Equation (15.3-1) is satisfied, payoff improvement may begin.

15.3.3 Payoff Improvement

Once the test in Equation (15.3-1) is satisfied, the $d\Psi_i$ vector may be expanded to include $d\Phi$, the payoff improvement.

The first payoff improvement is either an input quantity or can be estimated from an input of the expected final value of performance index Φ.

Subsequent program iterations will attempt to get further payoff improvement. If an asked-for $d\Phi$ is very large, control divergence or payoff divergence may occur or constraints may be violated. In the event that any of these occur, $d\Phi$ will be scaled down by $1/\sqrt{2}$ and the metric $(dP)^2$ will be halved.

At each new payoff improvement iteration, the predicted $d\Phi$ is calculated using Equation (12.2-7). The new $d\Phi$ is compared with the old value to see if special convergence acceleration is appropriate. If so, the $(dP)^2$ metric is scaled up to tend to give larger payoff improvements per iteration. This payoff improvement logic is programmed in subroutine PAYØ2.

When the predicted $d\Phi$ is smaller than an input minimum payoff improvement, the program will integrate the solution trajectory. (When a QL solution is flagged, the transformation of adjoints to Euler-Lagrange multipliers (Section 14) occurs before the solution trajectory integration.)

Section 16
CONTROL VECTOR SOLUTION – OPTIMAL CONTROL

This section treats the determination of the optimal steering angles α and ϕ in the QL trajectory module of PADS.

Sections 16.1 through 16.5 establish the definitions of various quantities and notational conventions. In Section 16.6, the calculus of variations is applied to the trajectory optimization problem to derive the necessary conditions for optimality. Among these conditions are the optimal control laws for ϕ and α which are discussed in greater detail in Sections 16.7 and 16.8. Finally, in Section 16.9 the effects of control and state variable inequality constraints are discussed.

16.1 THE STATE VECTOR IN THE QL MODULE OF PADS

If the method of quasi-linearization were amenable to problems with variable end-points, the state vector in the QL module of PADS would be the same, except for order, as it is in the steepest descent module. However, to circumvent the problem of variable end-points, QL makes use of a transformation that incorporates an additional state variable.

The transformation runs as follows: Let z denote the original state vector and suppose that z conforms to the differential constraint

$$\frac{dz}{dt} = g(t, z, U), \quad t_o \leq t \leq t_F \tag{16.1-1}$$

where t is the independent variable (time), U is the control vector, t_F is unknown, and g is a known vector function. Define

$$\tau = t_F - t_o \tag{16.1-2}$$

and take

$$x = (t - t_o)/\tau \tag{16.1-3}$$

It follows that $x(t_F) = 1$ and $x(t_o) = 0$, regardless of the value of t_F. Let us define a new state vector y* as

$$y^*(x) = z(t(x)), \quad 0 \le x \le 1. \tag{16.1-4}$$

Then, by the chain rule of differentiation,

$$\frac{dy^*}{dx} = \frac{dz}{dt}\frac{dt}{dx} = \tau\frac{dz}{dt}, \quad 0 \le x \le 1. \tag{16.1-5}$$

Finally, we construct the actual state vector that QL treats by appending the variable τ to the vector y*. Call this vector y. Then the quasitime derivative of y is well defined as

$$\frac{dy}{dx} = f(x, y, U), \quad 0 \le x \le 1 \tag{16.1-6}$$

where

$$\frac{dy_i}{dx} = \tau\frac{dz_i}{dt} \tag{16.1-7}$$

if y_i is a member of y* and

$$\frac{dy_i}{dx} = 0 \tag{16.1-8}$$

if $y_i = \tau$

As a consequence, the state vector in QL is defined as

$$y = (V, \gamma, \psi, h, \rho, \mu, m, \tau, Q)^T \qquad (16.1\text{-}9)$$

For a two-point problem — i. e., a one arc problem — the variable τ is simply the duration of the arc in seconds. In the next section, we will discuss the ramifications of multi-point problems.

16.2 DESCRIPTION OF THE MULTI-POINT PROBLEM

In the course of flying the trajectory, should anything happen to cause the right side of Equation (16.1-1) to be discontinuous, then we have a multipoint problem. The point at which the discontinuity occurs is called a corner point.

PADS recognizes four types of corner points: (1) branch points on the late side of which Equation (16.1-1) takes on more than one value; (2) end points which are the last points of branches; (3) initial points which are the first points of trajectories; and (4) intermediate points which are corner points that do not fall into any of the above categories. Since an intermediate point is not a branch point, the right side of Equation (16.1-1) will have exactly one value on the late side of an intermediate point.

If a problem has a branch point, it is called a branch problem. Presently, PADS can handle no more than one branch point in a trajectory, and the right side of Equation (16.1-1) can have only two values on the late side of the branch point.

The portions of the trajectory between the various corner points are termed subarcs. PADS assumes that all subarcs in a given trajectory are time-wise contiguous to some other subarc. In other words, there are no time gaps between any subarcs.

Figure 16.2-1 depicts a typical branch problem in PADS. The subarcs are numbered sequentially so that for a branch problem the first subarc of the second branch is assigned the next number after the last subarc of the first branch.

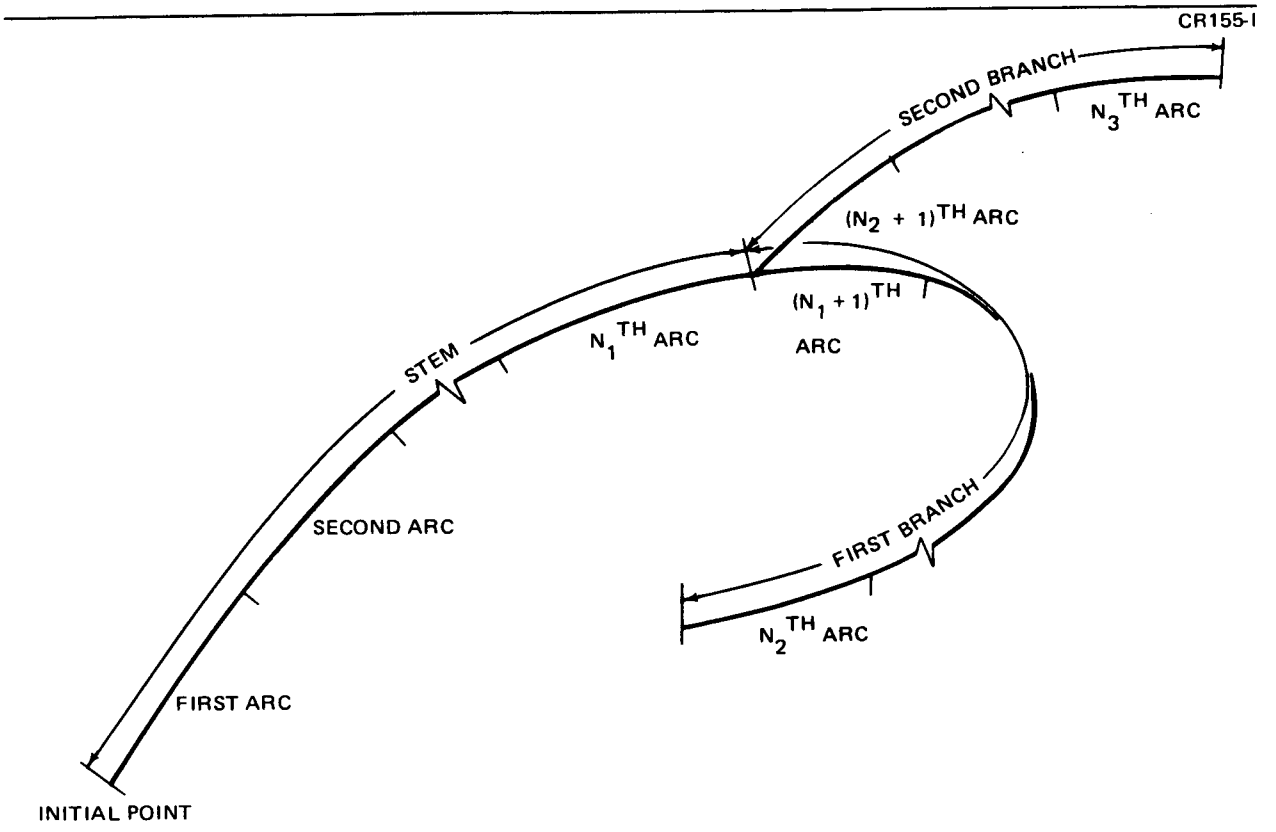


Figure 16.2-1. Branch Subarcs

As in the two-point problem discussed in Section 16. 1, the QL trajectory module transforms the time variable for the multipoint problem to the quasitime variable x. In this case, the transformation runs as follows: Let the corner points of the multipoint problem be ordered as follows:

$$t_o < t_1 < t_2 \cdots < t_{N_1} < \begin{cases} t_{N_1+1} < t_{N_1+2} \cdots < t_{N_2} & \text{(First Branch)} \\ t_{N_2+1} < t_{N_2+2} \cdots < t_{N_3} & \text{(Second Branch)} \end{cases} \tag{16.2-1}$$

For $0^+ \leq x \leq 1^-$, define

$$\tau(x) = t_1 - t_o \tag{16.2-2}$$

for $1^+ \leq x \leq 2^-$, define

$$\tau(x) = t_2 - t_1 \tag{16.2-3}$$

In general, for all I except $I = N_2+1$ and for $(I-1)^+ \le x \le I^-$, define

$$\tau(x) = t_I - t_{I-1} \qquad (16.2\text{-}4)$$

For $I = N_2+1$ (branch problems only) and $N_2^+ \le x \le (N_2+1)^-$, define

$$\tau(x) = t_{N_2+1} - t_{N_1} \qquad (16.2\text{-}5)$$

Hence, for $(I-1)^+ \le x \le I^-$, $\tau(x)$ is the duration of the I^{th} subarc.

16.3 THE EQUATIONS OF MOTION

In PADS, there are two different sets of equations of motion: the standard set which applies over most subarcs and a special set which applies only on vertical rise or pitch over subarcs. As a result, the equations of motion themselves are subarc-dependent, and, hence, for the multipoint problem, Equation (16.1-6) becomes

$$\frac{dy}{dx} = f_I(x, y, U), \quad (I-1)^+ \le x \le I^- \qquad (16.3\text{-}1)$$
(NLDRV)

where the subscript I on the right side of the equation indicates the subarc-dependency.

16.4 THE STATE BOUNDARY CONDITIONS

A full description of the various boundary conditions in PADS that the state can be made to satisfy is given in Section 5. It is sufficient to note here that all of the state boundary conditions fall into two broad categories: state initial conditions and state target conditions.

Let Ψ be the vector of all state boundary conditions for the entire multipoint problem. The QL module assumes that no state boundary condition involves the state at more than one corner point. Hence, we can partition Ψ into

$$\Psi = \begin{bmatrix} \psi_o \\ \psi_1 \\ \vdots \\ \psi_{N_1} \\ \psi_{N_1+1} \\ \vdots \\ \psi_{N_2} \\ \psi_{N_2+1} \\ \vdots \\ \psi_{N_3} \end{bmatrix} \begin{matrix} \} \text{ initial point} \\ \} \text{ intermediate points} \\ \\ \} \text{ branch point} \\ \} \text{ intermediate points} \\ \\ \} \text{ end-point of first branch} \\ \} \text{ intermediate points} \\ \\ \} \text{ end-point of second branch} \end{matrix} \qquad \begin{matrix} (16.4\text{-}1) \\ (\text{MAGIC}, \\ \text{BNDRY}) \end{matrix}$$

The vector ψ_o contains the state initial conditions at $x = 0^+$. ψ_1 contains the state initial conditions at $x = 1^+$ and the state target conditions at $x = 1^-$. In general, at the intermediate point I, ψ_I contains the state initial conditions at I^+ and the state target conditions at I^-. ψ_{N_1} contains the initial conditions at $N_2{}^+$, the initial conditions at $N_1{}^+$ and the target conditions at $N_1{}^-$. ψ_{N_2} contains the target conditions at $N_2{}^-$ and ψ_{N_3} contains the target conditions at $N_3{}^-$.

Since the QL state vector y has nine components, then the vector Ψ must have fewer than $18N_3$ components; otherwise, no optimization could occur.

16.5 THE CONTROL VECTOR

The control vector U in the QL module is defined as

$$U = (T, \delta_E, \alpha, \phi)^T \qquad (16.5\text{-}1)$$

There are several subvectors of U that have names of their own. The vector

$$u = (\alpha, \phi)^T \tag{16.5-2}$$

is termed the steering vector. On any given subarc, only the steering angles α and ϕ may be optimized. The vector

$$p = (T, \delta_E)^T \tag{16.5-3}$$

is called the propulsion vector. Neither component of p may be optimized on any subarc. The vector

$$w = (T, \delta_E, \alpha)^T \tag{16.5-4}$$

is called the in-plane control vector. As will be indicated later, the bank angle, ϕ, can always be determined independently of the vector w. This will be true both when ϕ is optimized and when it is not. As a consequence, the determination of the vector w will receive our greatest attention.

In any case, in order to evaluate Equation (16.3-1), it is necessary to determine the vector U at every point of the trajectory. Let K be the vector of algebraic constraints which may totally or only partially determine the vector U, i.e.,

$$K = K(x, y, U) = 0 \tag{16.5-5}$$

If K has four components, then U is completely determined and the vehicle is said to be undergoing non-optimal control. If K has less than four components, then U is only partially determined by K, and the vehicle is said to be undergoing optimal control. The minimum number of components K can have is two, in which case, both α and ϕ are free to be optimized.

Clearly, both the number of components and the equations assigned to the components of K are subarc-dependent. Hence, Equation (16.5-5) should be rewritten as

$$K_I = K_I(x, y, U) = 0 \qquad (16.5\text{-}6)$$
(CONTRL)

To completely determine U during optimal control, the indirect method of the calculus of variations is applied in the QL module of PADS. This will be discussed in detail in the following sections.

16.6 DERIVATION OF THE NECESSARY CONDITIONS FOR AN OPTIMAL SOLUTION OF THE MULTI-POINT PROBLEM

Multi-point trajectory optimization to which PADS addresses itself is an example of the problem of Mayer. The problem statement runs as follows:

We seek the state vector y(x) and control vector U(x) that cause the payoff function

$$\Phi = \Phi\,(y\,(N_3)) \qquad (16.6\text{-}1)$$
(BNDRY)

to experience a minimum subject to the differential constraints

$$\frac{dy}{dx} - f_I(x, y, U) = 0,\ 0 \le x \le N_3 \qquad (16.6\text{-}2)$$
(NLDRV)

and the algebraic constraints

$$K_I(x, y, U) = 0,\ 0 \le x \le N_3 \qquad (16.6\text{-}3)$$
(CONTRL)

and the fewer than $18N_3$ boundary conditions

$$\Psi = 0 \qquad (16.6\text{-}4)$$
(MAGIC, BNDRY)

Any valid target condition in the program can legally serve as a payoff. The fact that QL always minimizes the payoff should not concern the user, for if the maximization of the indicated payoff is desired, then QL minizes the additive inverse of the payoff. For example, suppose the maximization of $m(N_3)$ is desired. Then QL minimizes $-m(N_3)$.

There is one exceptional payoff function in the program that involves the state at more than the last point of the last branch. It is called the payload function and it is functionally dependent upon the booster and orbiter burnout masses; i. e.,

$$\Phi = \Phi\left(m\left(N_1^-\right),\ m\left(N_3^-\right)\right) \qquad \begin{matrix}(16.6\text{-}5)\\(\text{BNDRY})\end{matrix}$$

This payoff applies only for Phase I sizing problems and will be treated separately.

The approach to the problem stated above that the QL module of PADS employs is to adjoin the differential constraints and boundary conditions to the payoff to form the so-called augmented functional

$$J = \Phi + M \cdot \Psi + \sum_{I=1}^{N_3} \int_{I-1}^{I} \lambda \cdot \left(\frac{dy}{dx} - f_I\right) dx \qquad (16.6\text{-}6)$$

where M and λ are vectors of Lagrange multipliers. In QL, the λ vector is defined as

$$\lambda = \left(\lambda_V,\ \lambda_\gamma,\ \lambda_\psi,\ \lambda_\rho,\ \lambda_\mu,\ \lambda_m,\ \lambda_\tau,\ \lambda_Q\right)^T \qquad (16.6\text{-}7)$$

and is referred to as the costate vector.

Clearly, if y(x) and U(x) satisfy Equations (16. 6-2) through (16. 6-4), then Equation (16. 6-6) becomes

$$J = \Phi \qquad (16.6\text{-}8)$$

Hence, the object of the QL module is to determine the y(x) and U(x) that satisfy Equations (16.6-2) through (16.6-4) and cause Equation (16.6-6) to have a minimum value.

Before considering this problem, however, let us make use of the partition of Ψ introduced in Section 16.4 in Equation (16.6-6). We then get

$$J = \Phi + \mu_o \cdot \psi_o + \sum_{I=1}^{N_3} \mu_I \cdot \psi_I + \sum_{I=1}^{N_3} \int_{I-1}^{I} \lambda \cdot \left(\frac{dy}{dx} - f_I \right) dx \qquad (16.6\text{-}9)$$

where the vector M has been partitioned into the subvectors μ_o, μ_1, ..., μ_{N_3} in the same manner as Ψ.

The first necessary condition for the minimization of J is that

$$\delta J = 0 \qquad (16.6\text{-}10)$$

where δJ represents the total variation of J. Expanding Equation (16.6-10) we get

$$\begin{aligned}
\delta J = {} & \left[\partial \Phi / \partial y(N_3^-) \right] \delta y(N_3^-) + \mu_o \cdot \left[\partial \psi_o / \partial y(0^+) \right] \delta y(0^+) \\
& + \sum_{I \epsilon S} \mu_I \cdot \left[\partial \psi_I / \partial y(I^+) \;\vdots\; \partial \psi_I / \partial y(I^-) \right] \begin{pmatrix} \delta y(I^+) \\ \delta y(I^-) \end{pmatrix} \\
& + \mu_{N_1} \cdot \left[\partial \psi_{N_1} / \partial y(N_2^+) \;\vdots\; \partial \psi_{N_1} / \partial Y(N_1^+) \;\vdots\; \partial \psi_{N_1} / \partial y(N_1^-) \right] \begin{pmatrix} \delta y(N_2^+) \\ \delta y(N_1^+) \\ \delta y(N_1^-) \end{pmatrix} \\
& + \mu_{N_2} \cdot \left[\partial \psi N_2 / \partial y(N_2^-) \right] \delta y(N_2^-) + \mu_{N_3} \cdot \left[\partial \psi N_3 / \partial y(N_3^-) \right] \delta y(N_3^-) \\
& + \sum_{I=1}^{N_3} \delta \int_{I-1}^{I} \lambda \cdot \left(\frac{dy}{dx} - f_I \right) dx
\end{aligned} \qquad (16.6\text{-}11)$$

where

$$S = \{I : 1 \le I \le N_3 \text{ and } I \ne N_1, N_2, N_3\} \tag{16.6-12}$$

The last term of Equation (16.6-11) is the sum of the total variations of N_3 definite integrals. Let us consider only one of those variations — say, the I^{th} one. Furthermore, let us denote dy/dx by y'. Then, we have

$$\delta \int_{I-1}^{I} \lambda \cdot (y' - f_I)\, dx = \int_{I-1}^{I} \lambda \cdot \delta(y')\, dx - \int_{I-1}^{I} \lambda \cdot f_{I_y}\, \delta y\, dx - \int_{I-1}^{I} \lambda \cdot f_{I_U}\, \delta U\, dx \tag{16.6-13}$$

Noting that $\delta(y') = (\delta y)'$, the first term on the right side of Equation (16.6-13) can be integrated by parts to yield

$$D_I = \left[\lambda \cdot \delta y\right]_{I-1}^{I} - \int_{I-1}^{I} \frac{d\lambda}{dx}\, \delta y\, dx - \int_{I-1}^{I} \lambda \cdot f_{I_y}\, \delta y\, dx - \int_{I-1}^{I} \lambda \cdot f_{I_U}\, \delta U\, dx \tag{16.6-14}$$

where D_I denotes the expression on the left side of Equation (16.6-13).

Let the vector σ consist of those steering angles that are free to be optimized on the I^{th} subarc and let the vector q consist of the remainder of the U vector. Then

$$U = \left(\frac{q}{\sigma}\right) \tag{16.6-15}$$

As pointed out in Section 16.5, if K_I has four components, then σ is the empty vector and q = U. If K_I has two components, then σ = u and q = p. If K_I has three components, there are two possibilities depending on which steering angle is to be optimized. If ϕ (the bank angle) is to be optimized,

then $\sigma = (\phi)$ and $q = w$. If α is to be optimized, then $\sigma = (\alpha)$ and $q = (T, \delta_E, \phi)^T$.

In any case, because of Equation (16.6-3), there is an implicit function k so that

$$q = k(x, y, \sigma) \tag{16.6-16}$$

(see Reference 11) and, as a consequence, Equation (16.6-14) can be rewritten as

$$D_I = \left[\lambda \cdot \delta y\right]_{I-1}^{I} - \int_{I-1}^{I} \frac{d\lambda}{dx} \cdot \delta y \, dx - \int_{I-1}^{I} \lambda \cdot f_{I_y} \, \delta y \, dx$$
$$- \int_{I-1}^{I} \lambda \cdot f_{I_q} \, k_y \, \delta y \, dx - \int_{I-1}^{I} \lambda \cdot f_{I_q} \, k_\sigma \, \delta\sigma \, dx - \int_{I-1}^{I} \lambda \cdot f_{I_\sigma} \, \delta\sigma \, dx \tag{16.6-17}$$

After collecting the terms under the integrals in Equation (16.6-17) on δy and $\delta\sigma$, D_I becomes

$$D_I = \left[\lambda \cdot \delta y\right]_{I-1}^{I} - \int_{I-1}^{I} \left[\left(\frac{d\lambda}{dx}\right)^T + \lambda \cdot \left(f_{I_y} + f_{I_q} \, k_y\right)\right] \delta y \, dx$$
$$- \int_{I-1}^{I} \lambda \cdot \left(f_{I_\sigma} + f_{I_q} \, k_\sigma\right) \, \delta\sigma \, dx \tag{16.6-18}$$

Since δy and $\delta\sigma$ appear under the integral sign, both of the integrals in Equation (16.6-18) must vanish in order to satisfy Equation (16.6-10). Hence, by the

Fundamental Lemma of the calculus of variations (Reference 12) the Euler-Lagrange equations

$$\left(\frac{d\lambda}{dx}\right)^T + \lambda \cdot \left(f_{I_y} + f_{I_q} k_y\right) = 0, \quad (I-1)^+ \leq x \leq I^- \qquad \begin{matrix}(16.6\text{-}19)\\(\text{NLDRV})\end{matrix}$$

and

$$-\lambda \cdot \left(f_{I_\sigma} + f_{I_q} k_\sigma\right) = 0, \quad (I-1)^+ \leq x \leq I^- \qquad \begin{matrix}(16.6\text{-}20)\\(\text{CONTRL})\end{matrix}$$

must be satisfied at each point of the I^{th} subarc.

Assuming that Equations (16.6-19) and (16.6-20) are satisfied, then Equation (16.6-18) reduces to

$$D_I = \left[\lambda \cdot \delta y\right]_{I-1}^{I} = \left(\frac{\lambda(I^-)}{-\lambda(I-1^+)}\right) \cdot \left(\frac{\delta y(I^-)}{\delta y(I-1^+)}\right) \qquad (16.6\text{-}21)$$

and the last term on the right side of Equation (16.6-11) becomes

$$\sum_{I=1}^{N_3} \delta \int_{I-1}^{I} \lambda \cdot \left(\frac{dy}{dx} - f_I\right) dx = \sum_{I=1}^{N_3} \left(\frac{\lambda(I^-)}{-\lambda(I-1^+)}\right) \cdot \left(\frac{\delta y(I^-)}{\delta y(I-1^+)}\right) \qquad (16.6\text{-}22)$$

Still assuming that the Euler-Lagrange equations are satisfied on every subarc, Equation (16.6-10) in its expanded form becomes, after collecting terms at the corner points,

$$\delta J = 0 = E_o \, \delta y(0^+) + \sum_{I \epsilon S} E_I \left(\frac{\delta y(I^+)}{\delta y(I^-)}\right) + E_{N_1} \left(\begin{matrix}\delta y(N_2{}^+)\\ \delta y(N_1{}^+)\\ \delta y(N_1{}^-)\end{matrix}\right)$$

$$+ E_{N_2} \, \delta y(N_2{}^-) + E_{N_3} \, \delta y(N_3{}^-) \qquad (16.6\text{-}23)$$

where E_o, E_1, E_2, etc. are the row vectors

$$E_o = \mu_o \cdot \left[\partial\psi_o/\partial y(0^+)\right] - \lambda(0^+)^T \qquad \begin{matrix}(16.6\text{-}24)\\(\text{COSTAO})\end{matrix}$$

$$E_I = \mu_I \cdot \left[\partial\psi_I/\partial y(I^+) \;\vdots\; \partial\psi_I/\partial y(I^-)\right] - \left[\lambda(I^+)^T \;\vdots\; -\lambda(I^-)^T\right], I\epsilon S \qquad \begin{matrix}(16.6\text{-}25)\\(\text{COSTAI,}\\ \text{INTRPT})\end{matrix}$$

$$E_{N_1} = \mu_{N_1} \cdot \left[\partial\psi_{N_1}/\partial y(N_2^+) \;\vdots\; \partial\psi_{N_1}/\partial y(N_1^+) \;\vdots\; \partial\psi_{N_1}/\partial y(N_1^-)\right]$$

$$- \left[\lambda(N_2^+)^T \;\vdots\; \lambda(N_1^+)^T \;\vdots\; -\lambda(N_1^-)^T\right] \qquad \begin{matrix}(16.6\text{-}26)\\(\text{COSTAB,}\\ \text{BRANPT})\end{matrix}$$

$$E_{N_2} = \mu_{N_2} \cdot \left[\partial\psi_{N_2}/\partial y(N_2^-)\right] + \lambda(N_2^-)^T \qquad \begin{matrix}(16.6\text{-}27)\\(\text{ENDPT})\end{matrix}$$

$$E_{N_3} = \mu_{N_3} \cdot \left[\partial\psi_{N_3}/\partial y(N_3^-)\right] + \lambda(N_3^-)^T + \left[\partial\Phi/\partial y(N_3^-)\right] \qquad \begin{matrix}(16.6\text{-}28)\\(\text{ENDPT})\end{matrix}$$

Equation (16.6-23) is called the transversality equation. Since the state variations at the corner points are independent, the transverality equation will be satisfied only if E_o, E_1, ..., E_{N_3} all vanish. Hence, the problem is to determine the values of μ_o, μ_1, ..., μ_{N_3} that cause Equations (16.6-24) through (16.6-28) to vanish.

Toward this end, we note that Equations (16.6-24) through (16.6-28) are of the general form

$$\mu^T A + b = 0 \qquad (16.6\text{-}29)$$

where A is a matrix of partial derivatives and b is a row vector. For example, at the intermediate point I, μ in Equation (16.6-29) is μ_I,

$$b = -\left[\lambda (I^+)^T \;\vdots\; -\lambda (I^-)^T\right] \tag{16.6-30}$$

and

$$A = \left[\partial \psi_I / \partial y(I^+) \;\vdots\; \partial \psi_I / \partial y(I^-)\right] \tag{16.6-31}$$

The number of rows in μ and A equals the number of boundary conditions that apply at the corner point. The number of columns in A and b is 9 for initial and end points, 18 for intermediate points, and 27 for branch points. Let us denote the number of rows and columns in A by m and n, respectively. Presumably, the m rows of A are linearly independent row vectors.

Should m = n, the solution to Equation (16.6-29) is simply

$$\mu^T = -b\, A^{-1} \tag{16.6-32}$$

However, when m < n, μ is underdetermined, and, as a consequence, some additional boundary conditions must be added. These new conditions are called transversality conditions and they involve both the state and the costate.

The ontogeny of the transversality conditions runs as follows. Since the m rows of A are independent, they span a proper subspace of E^n. Take B to be any set of n - m row vectors so that A$\cup$B spans the entire space E^n. This generates an invertible n by n matrix.

$$A^* = \left[\frac{A}{B}\right] \tag{16.6-33}$$

compute

$$\left(\frac{\mu}{\theta}\right)^T = -b\, A^{*-1} \tag{16.6-34}$$

Then, if $\theta = 0$, it follows that

$$\left(\frac{\mu}{0}\right)^T \left[\frac{A}{B}\right] = \mu^T A + 0^T B = \mu^T A = -b \qquad (16.6\text{-}35)$$

Hence, to satisfy Equation (16.6-29) when $m < n$, we must determine $y(x)$ and $\lambda(x)$ so that θ, evaluated in Equation (16.6-34) vanishes.

In general, some transversality conditions will exist at each corner point. So we define the vector Θ to be the transversality conditions over the entire problem, i.e.

$$\Theta = \begin{bmatrix} \theta_o \\ \theta_1 \\ \vdots \\ \theta_{N_1} \\ \theta_{N_1+1} \\ \vdots \\ \theta_{N_2} \\ \theta_{N_2+1} \\ \vdots \\ \theta_{N_3} \end{bmatrix} \qquad \begin{matrix} (16.6\text{-}36) \\ (\text{MAGIC}, \\ \text{BNDRY}) \end{matrix}$$

For those problems in which the special payoff (Equation (16.6-5) applies, all we need do is add the partial of Φ with respect to $m(N_1^-)$ to the right side of Equation (16.6-26).

To summarize the development to this point, recall that we began with a set of differential constraints [Equation (16.6-2)]

$$\frac{dy}{dx} - f_I(x, y, U) = 0 \qquad \begin{matrix} (16.6\text{-}37) \\ (\text{NLDRV}) \end{matrix}$$

to which we have had to add the differential constraints [Equation (16.6-19)]

$$\left(\frac{d\lambda}{dx}\right)^T + \lambda \cdot \left(f_{I_y} + f_{I_q} k_y\right) = 0 \qquad \begin{matrix}(16.6\text{-}38)\\(\text{NLDRV})\end{matrix}$$

We also had a set of algebraic constraints [Equation (16.6-3)]

$$K_I(x, y, U) = 0 \qquad \begin{matrix}(16.6\text{-}39)\\(\text{CONTRL})\end{matrix}$$

and to these we have added the algebraic constraints [Equation (16.6-20)]

$$-\lambda \cdot \left(f_{I_\sigma} + f_{I_q} k_\sigma\right) = 0 \qquad \begin{matrix}(16.6\text{-}40)\\(\text{CONTRL})\end{matrix}$$

Finally, to the boundary conditions [Equation (16.6-4)]

$$\Psi = 0 \qquad \begin{matrix}(16.6\text{-}41)\\(\text{MAGIC, BNDRY})\end{matrix}$$

we have had to add the transversality conditions [Equation (16.6-36)]

$$\Theta = 0 \qquad \begin{matrix}(16.6\text{-}42)\\(\text{MAGIC, BNDRY})\end{matrix}$$

The terms k_y and k_σ in Equations (16.6-38) and (16.6-40) are, according to the implicit function theorem (Reference 11)

$$k_y = -\left[K_{I_q}\right]^{-1} K_{I_y} \qquad \begin{matrix}(16.6\text{-}43)\\(\text{ALGCON})\end{matrix}$$

and

$$k_\sigma = -\left[K_{I_q}\right]^{-1} K_{I_\sigma} \qquad \begin{matrix}(16.6\text{-}44)\\(\text{ALGCON})\end{matrix}$$

Thus, the calculus of variations has transformed our original optimization problem stated by Equations (16.6-1) through (16.6-4) into the multipoint boundary value problem stated by Equations (16.6-37) through (16.6-42).

16.7 THE OPTIMAL BANK ANGLE

Substituting Equation (16.6-44) into Equation (16.6-40) yields

$$-\lambda \cdot \left(f_{I_\sigma} - f_{I_q} \left[K_{I_q} \right]^{-1} K_{I_\sigma} \right) = 0 \qquad \begin{matrix} (16.7\text{-}1) \\ (\text{CONTRL}) \end{matrix}$$

If ϕ (the bank angle) is a component of σ, then the ϕ-component of Equation (16.7-1) is

$$-\lambda \cdot \left(f_{I_\phi} - f_{I_q} \left[K_{I_q} \right]^{-1} K_{I_\phi} \right) = 0 \qquad \begin{matrix} (16.7\text{-}2) \\ (\text{CONTRL}) \end{matrix}$$

However, as an examination of the various component candidates of K_I in Section 10 will verify, when ϕ is a component of σ, none of the candidates is explicitly dependent upon ϕ. In other words, when $\phi \epsilon \sigma$, $K_{I_\phi} = 0$. Hence, Equation (16.7-2) simplifies to

$$-\lambda \cdot f_{I_\phi} = -\lambda_\gamma \gamma'_\phi - \lambda_\psi \psi'_\phi = 0 \qquad (16.7\text{-}3)$$

Plugging the expressions for γ'_ϕ and ψ'_ϕ into Equation (16.7-3) and simplifying yields

$$\lambda_\gamma \sin\phi - \lambda_\psi \cos\phi / \cos\gamma = 0 \qquad (16.7\text{-}4)$$

This equation has two solutions which are supplementary. Take

$$C = \sqrt{\lambda_\psi^2 + \lambda_\gamma^2 \cos^2\gamma} \qquad (16.7\text{-}5)$$

Then

$$\sin\phi = \lambda_\psi / C \qquad \begin{matrix} (16.7\text{-}6) \\ (\text{CONTRL}) \end{matrix}$$

$$\cos \phi = \lambda_\gamma \cos \gamma / C \qquad \begin{matrix}(16.7\text{-}7)\\(\text{CONTRL})\end{matrix}$$

and

$$\sin \phi = -\lambda_\psi / C \qquad \begin{matrix}(16.7\text{-}8)\\(\text{CONTRL})\end{matrix}$$

$$\cos \phi = -\lambda_\gamma \cos \gamma / C \qquad \begin{matrix}(16.7\text{-}9)\\(\text{CONTRL})\end{matrix}$$

are solutions to Equation (16.7-4). One solution will place ϕ in the first or fourth quadrant and is therefore termed the belly-down solution. The other will place ϕ in the second or third quadrant and is consequently called the belly-up solution.

If both solutions are admissible, then we apply Pontryagin's maximum principle to guide us in the choice of the best solution (Reference 13). According to this principle, if the functional J in Equation (16.6-9) is to experience a local minimum, then the Hamiltonian function as defined by

$$H = \lambda \cdot f_I \qquad \begin{matrix}(16.7\text{-}10)\\(\text{CONTRL})\end{matrix}$$

must experience a maximum. Hence, our choice of ϕ will be the one which yields the largest value of H.

One other important observation is to be made about ϕ. The optimal bank angle is independent of the other components of U. As pointed out in Section 9.1, the same is true of the non-optimal bank angles. Hence, on any subarc, we can solve for the bank angle explicitly, and the problem of determining U therefore simplifies to determining the subvector w.

16.8 THE OPTIMAL ANGLE OF ATTACK

If α (the angle of attack) is a component of σ, then the α-component of Equation (16.7-1) is

$$-\lambda \cdot \left(f_{I_\alpha} - f_{I_q} \left[K_{I_q} \right]^{-1} K_{I_\alpha} \right) = 0 \qquad \begin{matrix} (16.8\text{-}1) \\ (\text{ALI}) \end{matrix}$$

In view of the observations made in the preceding section, Equation (16.8-1) simplifies to

$$-\lambda \cdot \left(f_{I_\alpha} - f_{I_p} \left[K_{I_p} \right]^{-1} K_{I_\alpha} \right) = 0 \qquad \begin{matrix} (16.8\text{-}2) \\ (\text{ALI}) \end{matrix}$$

regardless of whether or not ϕ is optimal. Of course, K_I should now be viewed as having only two components.

At this point, we must consider what might happen to Equation (16.8-2) when the total acceleration limit is encountered at some point in the I^{th} subarc while the vehicle is still in the atmosphere. Just prior to hitting the limit, Equation (16.8-2) is simply

$$-\lambda \cdot \left(f_\alpha - f_{\delta_E} \left[K_{\delta_E}^{(2)} \right]^{-1} K_\alpha^{(2)} \right) = 0 \qquad \begin{matrix} (16.8\text{-}3) \\ (\text{ALI}) \end{matrix}$$

where the subscript I has been dropped because it is understood and the superscript $^{(2)}$ indicates the second component of K. At the point where the limit is met, however, $K_\alpha^{(1)} \neq 0$, and Equation (16.8-2) becomes

$$-\lambda \cdot \left\{ f_\alpha - \left(f_T \,\vdots\, f_{\delta_E} \right) \begin{bmatrix} K_T^{(1)} & K_{\delta_E}^{(1)} \\ K_T^{(2)} & K_{\delta_E}^{(2)} \end{bmatrix}^{-1} \begin{pmatrix} K_\alpha^{(1)} \\ K_\alpha^{(2)} \end{pmatrix} \right\} = 0 \qquad (16.8\text{-}4)$$

Unfortunately, Equations (16.8-3) and (16.8-4) do not necessarily yield the same value of α at the point where the limit is met. A discontinuity of α, moreover, necessitates the introduction of another corner point. Worse yet, the value of T may exceed its maximum and, thereby require the imposition of the total acceleration limit as a control constraint on α.

To avoid these difficulties, Equation (16.8-3) is taken as the only form of the optimal control law for α.

16.9 CONTROL AND STATE VARIABLE INEQUALITY CONSTRAINTS

There are three first-order state variable inequality constraints and three control constraints. In both cases, the angle of attack is chosen so that the constraint value is satisfied. Also, in both cases, the assumed underlying unconstrained control equation is the optimal control law [Equation (16.8-3)]

In the case of a state variable inequality constraint (SVIC), the constrained portion of flight must begin on the late side of a corner point because this portion of flight begins by matching the constraint as a boundary condition (Reference 14). Thereafter, the angle of attack will follow the SVIC until the optimal α resulting from Equation (16.8-3) causes the time-rate of change of the SVIC to be negative. For example, suppose the SVIC is the dynamic pressure, q. At the corner point, the constraint is matched; i.e., $q = q_{max}$. Thereafter, if

$$\frac{dq}{dt}(\alpha_{opt}) > 0 \qquad (16.9\text{-}1)$$

then α is chosen so that

$$\frac{dq}{dt}(\alpha) = 0 \qquad (16.9\text{-}2)$$

but if

$$\frac{dq}{dt}(\alpha_{opt}) \leq 0 \qquad (16.9\text{-}3)$$

then α_{opt} is used.

Once off an SVIC, the program cannot go back on without the introduction of another corner point.

Ordinary control constraints, on the other hand, do not require a corner point and, consequently, the constraint can go on and off at will.

For both SVIC's and control constraints, at the point where the switch from constrained to optimal control occurs, it should be clear that Equation (16.6-37) is continuous. However, the continuity of Equation (16.6-38) is not as obvious. In order to prove continuity, we first observe that at the switch point, both the optimal and the constrained angle of attack satisfy

$$-\lambda \cdot \left(f_\alpha - f_p K_p^{-1} K_\alpha \right) = 0 \tag{16.9-4}$$

On the early side of the switch, Equation (16.6-38) is

$$\left(\frac{d\lambda}{dx}\right)^T = -\lambda \cdot \left\{ f_y - (f_p \mid f_\alpha) \left[\begin{array}{c|c} K_p & K_\alpha \\ \hline C_p & C_\alpha \end{array}\right]^{-1} \left(\begin{array}{c} K_y \\ \hline C_y \end{array}\right) \right\} \tag{16.9-5}$$

where C is the constraint. On the late side of the switch, Equation (16.6-38) is

$$\left(\frac{d\lambda}{dx}\right)^T = -\lambda \cdot \left(f_y - f_p K_p^{-1} K_y \right) \tag{16.9-6}$$

If Equations (16.9-5) and (16.9-6) are to be equal, then we must show that

$$\lambda \cdot f_p K_p^{-1} K_y = \lambda \cdot (f_p \mid f_\alpha) \left[\begin{array}{c|c} K_p & K_\alpha \\ \hline C_p & C_\alpha \end{array}\right]^{-1} \left(\begin{array}{c} K_y \\ \hline C_y \end{array}\right) \tag{16.9-7}$$

Since K_p and C_a are nonsingular, the inversion in Equation (16.9-7) is given by

$$\left[\begin{array}{c|c} K_p & K_a \\ \hline C_p & C_a \end{array}\right]^{-1} = \left[\begin{array}{c|c} \left(K_p - K_a C_a^{-1} C_p\right)^{-1} & -K_p^{-1} K_a \left(C_a - C_p K_p^{-1} K_a\right)^{-1} \\ \hline -C_a^{-1} C_p \left(K_p - K_a C_a^{-1} C_p\right)^{-1} & \left(C_a - C_p K_p^{-1} K_a\right)^{-1} \end{array}\right] \tag{16.9-8}$$

Multiplying the inverse by the row vector $\lambda \cdot (f_p \mid f_a)$ yields

$$\lambda \cdot (f_p \mid f_a) \left[\begin{array}{c|c} K_p & K_a \\ \hline C_p & C_a \end{array}\right]^{-1} = \left[\left(\lambda \cdot f_p - \lambda \cdot f_a C_a^{-1} C_p\right)\left(K_p - K_a C_a^{-1} C_p\right)^{-1} \mid \left(\lambda \cdot f_a - \lambda \cdot f_p K_p^{-1} K_a\right)\left(C_a - C_p K_p^{-1} K_a\right)^{-1}\right] \tag{16.9-9}$$

Substituting Equation (16.9-4) into Equation (16.9-9) yields

$$\begin{aligned} &\lambda \cdot (f_p \mid f_a) \left[\begin{array}{c|c} K_p & K_a \\ \hline C_p & C_a \end{array}\right]^{-1} \\ &\quad = \left[\left(\lambda \cdot f_p - \lambda \cdot f_p K_p^{-1} K_a C_a^{-1} C_p\right)\left(K_p - K_a C_a^{-1} C_p\right)^{-1} \mid 0\right] \\ &\quad = \left[\lambda \cdot f_p K_p^{-1}\left(K_p - K_a C_a^{-1} C_p\right)\left(K_p - K_a C_a^{-1} C_p\right)^{-1} \mid 0\right] \\ &\quad = \left[\lambda \cdot f_p K_p^{-1} \mid 0\right] \end{aligned} \tag{16.9-10}$$

Plugging Equation (16. 9-10) into Equation (16. 9-7) yields the desired result.

Hence, both the state and costate come off the constraint surface tangentially.

16. 10 SUMMARY

In view of the results of Sections 9. 1 and 16. 7, regardless of whether the bank angle is optimal or non-optimal, we can always evaluate it explicitly.

As a result, the problem of evaluating U reduces to solving for w. Section 9. 2 discussed this problem when α is non-optimal. Section 16. 8 shows how Equation (16. 8-3) fills in for the missing component of K_I when α is optimal. Hence, we can view K_I as always having three components regardless of whether α is optimal or non-optimal. However, if we do, we also must view K_I as being explicitly dependent upon the costate; i. e.

$$K_I = K_I(x, y, \lambda, w) = 0 \qquad (16.10\text{-}1)$$
(ALGCON)

Section 17

QUASILINEAR SOLUTION OF THE MULTI-POINT BOUNDARY VALUE PROBLEM

Recall that the calculus of variations transformed the original optimization problem into a multi-point boundary value problem whose solution will indirectly yield a local minimum of the payoff function (Section 16.6). The boundary value problem itself falls into the category of nonlinear first-order ordinary differential equations.

The numerical method employed by the QL trajectory module of PADS is known as quasi-linearization. This is an iterative technique which is actually an extension of the Newton-Raphson iteration to function spaces. Since a proof of the convergence of the iteration is beyond the scope of this document, the ensuing sections are intended as an exposition of how the method works in PADS rather than why.

A simple two-point problem is discussed in Section 17.1. A number of observations are made about the nature of the quasi-linear solution and the multi-point problem is then addressed in Section 17.2. The subsequent sections discuss the significant mathematical and numerical problems and techniques that are attendant on the solution of the multi-point problem.

17.1 THE TWO-POINT PROBLEM

Consider the system of first-order nonlinear ordinary differential equations

$$\frac{dY}{dx} = F(x, Y, w), \quad 0 \leq x \leq 1 \qquad (17.1\text{-}1)$$
(NLDRV)

where

$$Y^T = (y^T : \lambda^T)$$

and where

$$F^T = (f(x, y, w)^T \vdots -\lambda \cdot (f_y + f_q k_y)) \qquad (17.1\text{-}2)$$

and where, according to Equation (16.10-1), w is subject to the algebraic constraints

$$K(x, Y, w) = 0 \qquad (17.1\text{-}3)$$
(ALGCON)

together with the boundary conditions

$$(\Psi^T \vdots \Theta^T) = 0 \qquad (17.1\text{-}4)$$
(BNDRY)

Suppose Y is a solution to Equations (17.1-1) through (17.1-4), then provided F is twice continuously differentiable with respect to Y, Equation (17.1-1) can be written as a Taylor series

$$\frac{dY}{dx} = F(x, Z, W) + \left[\frac{\partial F}{\partial Y}\right]_{Y=Z} (Y - Z) + HOT \qquad (17.1\text{-}5)$$

where $Z = Z(x)$ is an element in the same function space as Y and W satisfies

$$K(x, Z, W) = 0 \qquad (17.1\text{-}6)$$

and HOT denotes a second-order remainder term.

The idea behind quasi-linearization is if Z can be chosen so that HOT is negligible, then the nonlinear system in Equation (17.1-1) can be approximated by the linear system

$$\frac{ds}{dx} = F(x, Z, W) + \left[\frac{\partial F}{\partial Y}\right]_{Y=Z} (s - Z) \qquad (17.1\text{-}7)$$
(LINDRV)

Because such a system is linear in s, it will have a solution of the form

$$s(x) = p(x) + \sum_{i=1}^{18} h_i(x)c_i \qquad (17.1\text{-}8)$$
(NOMINAL)

where p(x) is some particular solution of Equation (17.1-7)

$$\frac{dp}{dx} = F(x, Z, W) + \left[\frac{\partial F}{\partial Y}\right]_{Y=Z} (p - Z) \qquad (17.1\text{-}9)$$
(LINDRV)

$h_1(x)$, $h_2(x)$, ..., $h_{18}(x)$ are a set of linearly independent solutions of the homogeneous differential equation

$$\frac{dh_i}{dx} = \left[\frac{\partial F}{\partial Y}\right]_{Y=Z} h_i \qquad (17.1\text{-}10)$$
(LINDRV)

and $c_1, c_2, \ldots, c_{18}$ is any set of scalars that causes the equation

$$(\Psi(s)^T \vdots \Theta(s)^T) = 0 \qquad (17.1\text{-}11)$$
(BNDRY)

to be satisfied.

Since HOT has been neglected, however, s will not generally be a solution to Equation (17.1-1). As a result, an iterative process is employed wherein Z is replaced by s in Equation (17.1-7) and a new s is computed.

If this process converges; i.e., if for any positive ϵ, a positive integer M exists so that

$$\max_{0 \le x \le 1} \left|\left| s_m(x) - s_{m+1}(x) \right|\right| < \epsilon \qquad (17.1\text{-}12)$$

whenever m > M (m denotes iteration number), then it can be shown that it converges to a solution of Equation (17.1-1). Moreover, it converges at a rate that is quadratic (Reference 15).

Since the particular solution p(x) can be any solution of Equation (17.1-7), we establish the following advantageous conventions in choosing the initial value p(0). Some of the boundary conditions in $(\Psi^T : \Theta^T)$ will be initial conditions on the state or costate. For example,

$$m(0) - 50,000 \text{ slugs} = 0 \qquad (17.1\text{-}13)$$

For each such condition, the state or costate variable involved is said to be fixed (known). All state or costate variables that are not fixed are said to be free (unknown). For all fixed variables, we set the appropriate components of p(0) to the known values. Thus, continuing with our example, the seventh component of p(0) is set to 50,000. For all free variables, the appropriate components of p(0) will be set to the corresponding components of Z(0). For example, suppose $\lambda_V(0)$ is free, then

$$p_{10}(0) = Z_{10}(0) \qquad (17.1\text{-}14)$$

Of course, on the first QL iteration, Z(0) is the value of the initial arc at x = 0+. On subsequent iterations, however, as we have already noted

$$Z(0) = s_{m-1}(0) \qquad (17.1\text{-}15)$$

The homogeneous solutions $h_1(x), h_2(x), \ldots, h_{18}(x)$ must be independent. Hence, we use the following convention in choosing their initial values

$$h_i(0) = \delta_{ij} = \begin{cases} 1 \text{ if } i=j \\ 0 \text{ if } i\neq j \end{cases} \qquad (17.1\text{-}16) \text{ (SALVE)}$$

As a result of this convention, if H is the matrix whose columns are $h_i(x)$, i=1, ..., 18, then

$$H(0) = I \tag{17.1-17}$$

the identity matrix. Moreover,

$$s(0) = p(0) + H(0)c = p(0) + c \tag{17.1-18}$$

Since the values of the scalars $c_1, c_2, \ldots, c_{18}$ are chosen so that Equation (17.1-11) holds and since, by convention, p(0) is chosen so that it satisfies the initial conditions in Equation (17.1-4), it is clear from Equation (17.1-18) that c_i is trivially zero if Y_i is fixed. Hence, we can throw out those homogeneous solutions that correspond to fixed states.

Thus the following becomes the actual convention for the homogeneous solutions: For each free variable Y_j introduce a homogeneous solution $h_i(x)$ whose initial value is

$$h_i(0) = \delta_{ij} \tag{17.1-19}$$

In actuality, then, the matrix H(x) will have 18 rows and up to 18 columns, depending on the number of free variables, n.

Equation (17.1-18) also conveys the meaning of the remaining c's. They are the necessary perturbations to the initial values of the free variables to cause s(1) to satisfy the target conditions in Equation (17.1-4). Differentiating Equation (17.1-8) by a particular c_i indicates the meaning of the homogeneous solutions.

$$\frac{\partial s(x)}{\partial c_i} = h_i(x) \tag{17.1-20}$$

These solutions represent the sensitivity of s(x) to a unit perturbation at the initial point of the corresponding free variable.

In view of Equation (17.1-15), it is clear that one necessary condition of convergence in the sense of Equation (17.1-12) is that

$$\lim_{\epsilon \to 0} c = 0 \qquad (17.1\text{-}21)$$

Hence, as an alternate definition of convergence, we use

$$\sum_{i=1}^{n} |c_i| = 0 \qquad (17.1\text{-}22)$$

17.2 THE MULTI-POINT PROBLEM

For a multi-point problem Equations (17.1-1) through (17.1-3) become

$$\frac{dY}{dx} = F_I(x, Y, w), \quad I - 1^+ \leq x \leq I^-, \quad I = 1, 2, \ldots, N_3 \qquad (17.2\text{-}1)$$
(NLDRV)

$$F_I^T = \left(f_I(x, y, w)^T \vdots - \lambda \cdot \left(f_{I_y} + f_{I_q} k_y \right) \right) \qquad (17.2\text{-}2)$$

and

$$K_I(x, Y, w) = 0 \qquad (17.2\text{-}3)$$
(ALGCON)

respectively.

More important, the following conventions are adopted for the particular and homogeneous solutions.

If a variable is continuous across a corner point; i.e., if

$$Y_i(I^+) - Y_i(I^-) = 0 \qquad (17.2.4)$$

is one of the boundary conditions in Equation (17.1-4), then the corresponding component of the particular solution also goes across the corner continuously,

$$p_i(I^+) = p_i(I^-) \qquad (17.2\text{-}5)$$
(SALVE)

If the mass state variable experiences a discontinuity of known or computable magnitude

$$m(I^+) - m(I^-) + 5000 \text{ slugs} = 0 \qquad (17.2\text{-}6)$$

for example, then the seventh component of the particular solution experiences the same discontinuity

$$p_7(I^+) = p_7(I^-) - 5000 \text{ slugs} \qquad (17.2\text{-}7)$$

If a variable Y_j is free on the late side of a corner point, then set

$$p_j(I^+) = Z_j(I^+) \qquad (17.2\text{-}8)$$
(SALVE)

and introduce a new homogeneous solution $h_i(x)$ whose initial value at $x = I^+$ is given by

$$h_i(I^+) = \delta_{ij} \qquad (17.2\text{-}9)$$
(SALVE)

This is perfectly legal as long as we remember that the scalar for this homogeneous solution cannot be perturbed to satisfy target conditions prior to $x = I^+$.

(Because of the last rule, the number of homogeneous solutions increases monotonically as we progress from one subarc to the next.)

If a variable is fixed on the late side of a corner point; i.e. if

$$Y_i(I^+) - \begin{pmatrix}\text{known}\\ \text{value}\end{pmatrix} = 0 \qquad (17.2\text{-}10)$$

is a boundary condition in Equation (17.1-4), then set

$$p_i\,(I^+) = \begin{pmatrix}\text{known}\\ \text{value}\end{pmatrix} \qquad (17.2\text{-}11)\ (\text{SALVE})$$

and zero out the i^{th} row of the matrix $H(I^+)$. This is valid because the i^{th} variable at $x = I^+$ has become insensitive to perturbations of free variables prior to $x = I^+$.

On a branch problem, if

$$Y_i\left(N_2^+\right) - Y_i\left(N_1^-\right) = 0 \qquad (17.2\text{-}12)$$

is a component of Equation (17.1-4), set

$$p_i\left(N_2^+\right) = p_i\left(N_1^-\right) \qquad (17.2\text{-}13)\ (\text{SALVE})$$

If

$$m\left(N_2^+\right) - m\left(N_1^-\right) + 5000\ \text{slugs} = 0 \qquad (17.2\text{-}14)$$

is a component of Equation (17.1-4), set

$$p_7\left(N_2^+\right) = p_7\left(N_1^-\right) - 5000\ \text{slugs} \qquad (17.2\text{-}15)$$

If the mass is distributed between the two branches; i.e., if

$$m\left(N_2^+\right) + m\left(N_1^+\right) - m\left(N_1^-\right) = 0 \qquad (17.2\text{-}16)$$

is a component of Equation (17.1-4) set

$$p_7\left(N_2^+\right) = m\left(N_1^-\right) - m\left(N_1^+\right) \qquad (17.2\text{-}17)$$

If a costate is distributed between the branches; i.e., if there is a transversality condition such as

$$\lambda_V\left(N_2^+\right) + \lambda_V\left(N_1^+\right) - \lambda_V\left(N_1^-\right) = 0 \qquad (17.2\text{-}18)$$

in Equation (17.1-4), then distribute the corresponding component of p. For example, Equation (17.2-18) would result in

$$p_{10}\left(N_2^+\right) = p_{10}\left(N_1^-\right) - p_{10}\left(N_1^+\right) \qquad \begin{matrix}(17.2\text{-}19)\\(\text{SALVE})\end{matrix}$$

The distribution of a costate results from the state going across the branch point continuously to both branches.

The above conventions are chosen because they force

$$s(x) = p(x) + H(x)c \qquad (17.2\text{-}20)$$

to automatically satisfy all of the initial conditions in Equation (17.1-4). As a result, the determination of the c's is based solely on matching the target conditions in Equation (17.1-4).

17.3 RECOGNIZING THE INITIAL CONDITIONS ON THE COSTATE

Since the transversality conditions Θ are derived numerically, the program must have some means of recognizing those transversality conditions that represent initial conditions on the costate.

Let us first consider the initial point $x = 0^+$. At this point, Equation (16.6-34) becomes

$$\left(\frac{\mu_o}{\theta_o}\right)^T = \lambda(0+)^T A_o^{*-1} \qquad \begin{matrix}(17.3\text{-}1)\\(\text{COSTAO})\end{matrix}$$

Suppose θ_{o_i} is the j^{th} component of $\left(\mu_o^T : \theta_o^T\right)$. Then

$$\theta_{o_i} = \lambda(0^+)^T \left[A_o^{*-1}\right]^j \qquad \begin{matrix}(17.3\text{-}2)\\(\text{COSTAO})\end{matrix}$$

where

$\left[A_o^{*-1}\right]^j$ is the j^{th} column of A_o^{*-1}. If

$$\left[A_o^{*-1}\right]^j = \delta_{k\ell} = \begin{cases} 0 \text{ if } k \neq \ell \\ \\ 1 \text{ if } k = \ell \end{cases} \qquad \begin{matrix} (17.3\text{-}3) \\ (\text{COSTAD}) \end{matrix}$$

for some ℓ, $1 \leq \ell \leq 9$, then θ_{o_i} is of the form

$$\theta_{o_i} = \lambda_\ell(0^+) \qquad (17.3\text{-}4)$$

and the transversality condition is

$$\lambda_\ell(0^+) = 0 \qquad (17.3\text{-}5)$$

In other words, λ_ℓ is fixed at the initial point.

In fact, because the state variables at the initial point can only be fixed or free, the same will be the case for the costate variables.

Next, consider the intermediate point I. At this point Equation (16.6-34) becomes

$$\begin{pmatrix} \mu_I \\ \hline \theta_I \end{pmatrix}^T = \left[\lambda(I^+)^T \vdots -\lambda(I^-)^T\right] A_I^{*-1} \qquad \begin{matrix} (17.3\text{-}6) \\ (\text{COSTAI}) \end{matrix}$$

Suppose θ_{I_i} is the j^{th} component of $\left(\mu_I^T \vdots \theta_I^T\right)$. Then

$$\theta_{I_i} = \left[\lambda(I^+)^T \vdots -\lambda(I^-)^T\right]\left[A_I^{*-1}\right]^j \qquad \begin{matrix} (17.3\text{-}7) \\ (\text{COSTAI}) \end{matrix}$$

Let us denote the top nine entries of

$$\left[A_I^{*-1}\right]^j$$

by S and the bottom nine by T. Then, if T = 0 and $S = \delta_{k\ell}$ for some ℓ, $1 \le \ell \le 9$,

$$\theta_{I_i} = \lambda_\ell(I^+) \tag{17.3-8}$$

represents the fixed initial condition on λ_ℓ. If $S = \delta_{k\ell}$ for some ℓ, $1 \le \ell \le 9$, and T = S, then

$$\theta_{I_i} = \lambda_\ell(I^+) - \lambda_\ell(I^-) \tag{17.3-9}$$

represents the continuous initial condition on λ_ℓ.

Finally, consider the branch point N_1. At this point, Equation (16.6-34) becomes

$$\begin{pmatrix} \mu_{N_1} \\ \hline \theta_{N_1} \end{pmatrix}^T = \left[\lambda\left(N_2^+\right)^T \vdots \lambda\left(N_1^+\right)^T \vdots -\lambda\left(N_1^-\right)^T\right] A_{N_1}^{*-1} \tag{17.3-10 (COSTAB)}$$

Suppose θ_i is the j^{th} component $\left(\mu_{N1}^T \vdots \theta_{N1}^T\right)$. Then

$$\theta_i = \left[\lambda\left(N_2^+\right)^T \vdots \lambda\left(N_1^+\right)^T \vdots -\lambda\left(N_1^-\right)^T\right]\left[A_{N_1}^{*-1}\right]^j \tag{17.3-11 (COSTAB)}$$

Let us denote the top nine entries of

$$\left[A_{N_1}^{*-1}\right]^j$$

by R, the middle nine by S, and the bottom nine by T. Then, if T = S = 0 and $R = \delta_{k\ell}$ for some ℓ, $1 \leq \ell \leq 9$,

$$\theta_i = \lambda_\ell\left(N_2^+\right) \tag{17.3-12}$$

represents the fixed initial condition on $\lambda_\ell\left(N_2^+\right)$. If T = R = 0 and $S = \delta_{k\ell}$,

$$\theta_i = \lambda_\ell\left(N_1^+\right) \tag{17.3-13}$$

represents the fixed initial condition on $\lambda_\ell\left(N_1^+\right)$. If S = 0 and $R = T = \delta_{k\ell}$, then

$$\theta_i = \lambda_\ell\left(N_2^+\right) - \lambda_\ell\left(N_1^-\right) \tag{17.3-14}$$

represents the continuity initial condition on λ_ℓ between the trunk and the second branch. If R = 0 and $S = T = \delta_{k\ell}$, then

$$\theta_i = \lambda_\ell\left(N_1^+\right) - \lambda_\ell\left(N_1^-\right) \tag{17.3-15}$$

represents the continuity initial condition on λ_ℓ between the trunk and the first branch. Finally, if $R = S = T = \delta_{k\ell}$, then

$$\theta_i = \lambda_\ell\left(N_2^+\right) + \lambda_\ell\left(N_1^+\right) - \lambda_\ell\left(N_1^-\right) \tag{17.3-16}$$

represents the costate distribution initial condition on λ_ℓ.

The fixed, continuous, and costate distribution conditions are the only transversality conditions recognized as initial conditions on the costate. All other transversality conditions are treated as costate target conditions.

17.4 SOLVING FOR THE C'S

Having integrated the particular and homogeneous solutions forward to the end of the trajectory, we must then determine the values of the scalars $c_1, c_2, \ldots, c_n$ that cause the target conditions in Equation (17.1-11) to

be satisfied. The method employed is the well-known Newton-Raphson interation for systems of equations.

Let us assume that Equation (17.1-11) has been reduced to target conditions only. Further, let us take the value 0 as an initial guess for the vector c. Define

$$\Delta c^{(i)} = c^{(i+1)} - c^{(i)} \tag{17.4-1}$$

where the superscript indicates the iteration number, and

$$c^{(0)} = \text{initial guess} = 0 \tag{17.4-2}$$

According to the method of Newton-Raphson, the increment in Equation (17.4-1) is given by

$$\Delta c^{(i)} = -\left[\frac{\partial \Psi / \partial c}{\partial \Theta / \partial c}\right]^{-1}_{c=c^{(i)}} \begin{pmatrix} \Psi (c^{(i)}) \\ \hline \Theta (c^{(i)}) \end{pmatrix} \tag{17.4-3 (NEWCS)}$$

Clearly, as the vector of target misses on the right side of Equation (17.4-3) approaches zero, so will the increment $\Delta c^{(i)}$ provided, of course, that the matrix of partial derivatives is non-singular.

To see how the target misses and partials in Equation (17.4-3) are actually evaluated, consider the target conditions Ψ_I at the corner point x = I. Suppose that prior to the point x = I, a total of m_I homogeneous solutions has been introduced. Let

$$c_I = \left(c_1, c_2, \ldots, c_{m_I}\right)^T \tag{17.4-4}$$

be the vector of multipliers for these homogeneous solutions. In general, of course, the vector

$$c = \left(c_1, c_2, \ldots, c_n\right)^T \tag{17.4-5}$$

of multipliers for all of the homogeneous solutions in the problem will have more components than c_I. Indeed

$$c_1 \subset c_2 \subset \ldots \subset c_I \subset \ldots \subset c_{N_3} \tag{17.4-6}$$

If we let

$$H_I(x) = \left[h_1(x) \vdots h_2(x) \vdots \cdots \vdots h_{m_I}(x) \right] \tag{17.4-7}$$

then

$$s(I^-) = p(I^-) + H_I(I^-)\, c_I \tag{17.4-8}$$

and, by the chain rule of differentiation,

$$\begin{aligned} \partial \Psi_I / \partial c_I &= (\partial \Psi_I / \partial s(I^-))\, (\partial s(I^-) / \partial c_I) \\ &= (\partial \Psi_I / \partial s(I^-))\, H_I(I^-) \end{aligned} \tag{17.4-9}$$

(INTRPT, BRANPT, ENDPT)

Of course, the partial derivatives of Ψ_I with respect to those c's in Equation (17.4-5) that are not in Equation (17.4-4) are zero.

Since the transversality conditions Θ_I at x = I are numerically derived, their partials with respect to the c's are numerically approximated by divided forward differences. For example, if c_i is one of the components of the vector c_I, then

$$\partial \theta_I / \partial c_i \approx (\theta_I(s(c_I + \Delta c_i) - \theta_I(s(c_I)))/\Delta c_i \tag{17.4-10}$$

(INTRPT, BRANPT, ENDPT)

Of course,

$$s(c_I + \Delta c_i) = s(c_I) + \Delta c_i h_i(I^-) \tag{17.4-11}$$

Some of the transversality conditions in θ_I may involve both sides of the corner point. Hence, the partials with respect to the c's introduced on the late side of the corner point must also be approximated by divided differences.

17.5 EVALUATION OF THE SYSTEM JACOBIAN

In order to numerically integrate Equations (17.1-9) and (17.1-10), we must evaluate the system Jacobian

$$\left[\frac{\partial F}{\partial Y}\right]_{Y=Z}$$

Recall that

$$Y^T = (y^T \,\vdots\, \lambda^T) \tag{17.5-1}$$

and

$$F^T = \left(f^T \,\vdots\, -\lambda \cdot (f_y + f_q k_y) \right) \tag{17.5-2}$$

Let

$$y' = f \tag{17.5-3}$$

and

$$\lambda'^T = -\lambda \cdot (f_y + f_q k_y) \tag{17.5-4}$$

Then

$$\left[\frac{\partial F}{\partial Y}\right]_{Y=Z} = \left[\frac{\partial y'}{\partial \lambda'} / \frac{\partial y}{\partial y} \,\vdots\, \frac{\partial y'}{\partial \lambda'} / \frac{\partial \lambda}{\partial \lambda}\right]_{Y=Z} \tag{17.5-5 (NLDRV)}$$

Naturally, the first step in evaluating the Jacobian is to solve Equation (17.1-3) for w. This problem has already been discussed for nonoptimal control modes in Section 10.2. Nevertheless, let us see what the four submatrices turn out to be when both α and ϕ are non-optimal.

When α is non-optimal, $q = w$. Hence

$$\frac{\partial y'}{\partial y} = f_y + f_w k_y + f_\phi \phi_y \qquad \begin{matrix}(17.5\text{-}6)\\(\text{NLDRV})\end{matrix}$$

As we noted in Section 10.1, there are only two non-optimal bank angle modes. For the mode in which

$$\phi = 0 \qquad (17.5\text{-}7)$$

the term ϕ_y vanishes. For the vertical rise or pitchover mode, the term f_ϕ vanishes. Hence, Equation (17.5-6) simplifies to

$$\frac{\partial y'}{\partial y} = f_y + f_w k_y \qquad \begin{matrix}(17.5\text{-}8)\\(\text{NLDRV})\end{matrix}$$

Since $q = w$, Equation (16.6-16) becomes

$$w = k(x, y) \qquad (17.5\text{-}9)$$

As a result, the term k_y in Equation (17.5-8) can be viewed as

$$k_y = \frac{\partial w}{\partial y} \qquad (17.5\text{-}10)$$

According to Equation (16.6-43)

$$\frac{\partial w}{\partial y} = -\left[K_w\right]^{-1} (K_y + K_\phi \phi_y)^* \qquad \begin{matrix}(17.5\text{-}11)\\(\text{ALGCON})\end{matrix}$$

When α and ϕ are non-optimal, Equation (17.5-4) becomes

$$\lambda'^T = -\lambda \cdot \left(f_y + f_w \frac{\partial w}{\partial y}\right) = -\lambda \cdot \frac{\partial y'}{\partial y} \qquad \begin{matrix}(17.5\text{-}12)\\(\text{NLDRV})\end{matrix}$$

---

*Recall that ϕ is an explicit function of y.

A number of shortcuts become apparent. First, compute $\partial y'/\partial y$ by means of Equations (17.5-8) and (17.5-11). Then compute λ' by means of Equation (17.5-12). Since w and ϕ do not depend on λ,

$$\frac{\partial y'}{\partial \lambda} = 0 \qquad \begin{matrix}(17.5\text{-}13)\\(\text{NLDRV})\end{matrix}$$

and

$$\frac{\partial \lambda'}{\partial \lambda} = -\left[\frac{\partial y'}{\partial y}\right]^T \qquad \begin{matrix}(17.5\text{-}14)\\(\text{NLDRV})\end{matrix}$$

Moreover, the matrix $\partial\lambda'/\partial y$ is symmetric. Observe first that

$$\left[\frac{\partial \lambda'}{\partial y}\right]_{ij} = \frac{\partial \lambda' i}{\partial y_j} = \frac{\partial}{\partial y_j}\left(-\frac{\partial y'}{\partial y_i}\right)\cdot\lambda = -\frac{\partial^2 y'}{\partial y_j \partial y_i}\cdot\lambda \qquad (17.5\text{-}15)$$

Now

$$\frac{\partial y'}{\partial y_i} = f_{y_i} + f_w \frac{\partial w}{\partial y_i} \qquad (17.5\text{-}16)$$

and, consequently,

$$\begin{aligned}\frac{\partial^2 y'}{\partial y_j \partial y_i} &= \frac{\partial}{\partial y_j}\left(f_{y_i} + f_w \frac{\partial w}{\partial y_i}\right)\\ &= \frac{\partial f_{y_i}}{\partial y_j} + \frac{\partial f_w}{\partial y_j}\frac{\partial w}{\partial y_i} + f_w \frac{\partial^2 w}{\partial y_j \partial y_i}\\ &= f_{y_i y_j} + f_w \frac{\partial w}{\partial y_j} + \left(f_{w y_j} + f_{w w}\frac{\partial w}{\partial y_j}\right)\frac{\partial w}{\partial y_i} + f_w \frac{\partial^2 w}{\partial y_j \partial y_i}\end{aligned} \qquad (17.5\text{-}17)$$

The right side of Equation (17.5-17) is symmetric with respect to y_i and y_j. Hence

E

$$\frac{\partial^2 y'}{\partial y_j \partial y_i} = \frac{\partial^2 y'}{\partial y_j \partial y_i} \tag{17.5-18}$$

which implies, by Equation (17.5-15), that

$$\left[\frac{\partial \lambda'}{\partial y}\right]_{ij} = \left[\frac{\partial \lambda'}{\partial y}\right]_{ji} \tag{17.5-19}$$

So we need only evaluate the upper triangular portion of $\partial \lambda'/\partial y$.

Of course, we still must evaluate the terms $\partial^2 w/\partial y_j \partial y_i$ for $j \geq i$. This evaluation has already been discussed in Section 10.3.

If ϕ is optimal and α is non-optimal, Equation (17.5-6) does not reduce to Equation (17.5-8). Of course, Equation (17.5-4) still holds, but Equations (17.5-12) through (17.5-14) and (17.5-19) no longer hold. In this case

$$\frac{\partial y'}{\partial \lambda} = f_\phi \phi_\lambda \tag{17.5-20 (NLDRV)}$$

and

$$\frac{\partial \lambda'_i}{\partial \lambda_j} = -\left(f_{y_i} + f_w \frac{\partial w}{\partial y_i}\right) \cdot \delta_{ij} - \left(f_{y_i \phi}\, \phi_{\lambda_j} + f_{w\phi}\, \phi_{\lambda_j} \frac{\partial w}{\partial y_j}\right) \cdot \lambda \tag{17.5-21 (NLDRV)}$$

Of course, $\partial \lambda'/\partial y$ is no longer symmetric.

$$\frac{\partial \lambda'_i}{\partial y_j} = -\left[f_{y_i y_j} + f_{w y_i} \frac{\partial w}{\partial y_j} + f_{\phi y_i}\, \phi_{y_j} + \left(f_{w y_j} + f_{w w} \frac{\partial w}{\partial y_j} + f_{w_\phi}\, \phi_{y_j}\right) \frac{\partial w}{\partial y_i} + f_w \frac{\partial^2 w}{\partial y_j \partial y_i}\right] \cdot \lambda \tag{17.5-22}$$

If both ϕ and α are optimal, then q = p. Let K* be the first two components of K in Equation (17.1-3). In other words

$$K^* = \begin{pmatrix} K^{(1)} \\ K^{(2)} \end{pmatrix} \qquad (17.5\text{-}23)$$

The third component of K is given by Equation (16.8-3). Equation (17.5-4) becomes

$$\lambda'^T = -\lambda \cdot (f_y + f_p p_y) \qquad (17.5\text{-}24)$$
(NLDRV)

where

$$p_y = -\left[K^*_p\right]^{-1} K^*_y \qquad (17.5\text{-}25)$$
(ALGCON)

In this case, the matrix $\partial y'/\partial y$ is still given by

$$\frac{\partial y'}{\partial y} = f_y + f_w \frac{\partial w}{\partial y} + f_\phi \phi_y \qquad (17.5\text{-}26)$$
(NLDRV)

as in Equation (17.5-6), and the matrix $\partial w/\partial y$ is still given by

$$\frac{\partial w}{\partial y} = -K_w^{-1}(K_y + K_\phi \phi_y) \qquad (17.5\text{-}27)$$
(ALGCON)

as in Equation (17.5-11). However, the matrix $\partial y'/\partial \lambda$ is now given by

$$\frac{\partial y'}{\partial \lambda} = f_w \frac{\partial w}{\partial \lambda} + f_\phi \phi_\lambda \qquad (17.5\text{-}28)$$
(NLDRV)

where

$$\frac{\partial w}{\partial \lambda} = -K_w^{-1}(K_\lambda + K_\phi \phi_\lambda) \qquad (17.5\text{-}29)$$
(ALGCON)

Moreover, the matrix $\partial\lambda'/\partial\lambda$ is given, element-by-element, by

$$\frac{\partial\lambda'_i}{\partial\lambda_j} = -\left(f_{y_i} + f_p p_{y_i}\right)\cdot\delta_{ij} - \left(f_{y_i u}\, u_{\lambda_j} + f_{pu}\, u_{\lambda_j} p_{y_i}\right)\cdot\lambda \qquad \begin{matrix}(17.5\text{-}30)\\(\text{NLDRV})\end{matrix}$$

where, of course, $u = (\alpha, \phi)$. The matrix $\partial\lambda'/\partial y$ is rather complicated and, needless to say, asymmetric. It is given, term-by-term, by

$$\frac{\partial\lambda'_i}{\partial y_j} = -\left(\frac{\partial f_i}{\partial y_j} + \frac{\partial f_p}{\partial y_j} p_{y_i} + f_p \frac{\partial p_{y_i}}{\partial y_j}\right)\cdot\lambda$$

$$= -\left[f_{y_i y_j} + f_{y_i w}\frac{\partial w}{\partial y_j} + f_{y_i\phi}\,\phi_{y_j} + \left(f_{py_j} + f_{pw}\frac{\partial w}{\partial y_j}\right) p_{y_i} + f_p \frac{\partial p_i}{\partial y_j}\right]\cdot\lambda$$

$$\begin{matrix}(17.5\text{-}31)\\(\text{NLDRV})\end{matrix}$$

where

$$\frac{\partial p_{y_i}}{\partial y_j} = \frac{\partial}{\partial y_j}\left(-\left[K^*_p\right]^{-1} K^*_{y_i}\right)$$

$$= -\left\{\left(\frac{\partial}{\partial y_j}\left[K^*_p\right]^{-1} + \left[K^*_p\right]^{-1}\frac{\partial K^*_{y_i}}{\partial y_j}\right\}\cdot\lambda\right.$$

$$= \left[K^*_p\right]^{-1}\left(\frac{\partial K^*_p}{\partial y_j}\left[K^*_p\right]^{-1} K^*_{y_i} - \frac{\partial K^*_{y_i}}{\partial y_j}\right)\cdot\lambda$$

$$= \left[K^*_p\right]^{-1}\left\{\left(K^*_{py_j} + K^*_{pw}\frac{\partial w}{\partial y_j}\right)\left[K^*_p\right]^{-1} K^*_{y_i} - K^*_{y_i y_j} - K^*_{y_{iw}}\frac{\partial w}{\partial y_j}\right\}$$

$$\begin{matrix}(17.5\text{-}32)\\(\text{ALGCON})\end{matrix}$$

In case α is optimal and ϕ is non-optimal (i.e., $\phi = 0$), then Equations (17.5-24) through (17.5-32) still apply, but the terms ϕ_y and ϕ_λ vanish.

17.6 METHODS OF INTEGRATION

Another major step in the procedure described in Sections 17.1 and 2 is the integration of Equations (17.1-9) and (17.1-10). This is accomplished by means of a fourth-order Adams-Moulton scheme with a standard fourth-order Runge-Kutta starting procedure.

Let F(x, s) denote the right side of Equation (17.1-9) or (17.1-10). Then, for the Runge-Kutta starting procedure, we have

$$s(x+h) = s(x) + \frac{h}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right] \qquad \begin{matrix}(17.6\text{-}1)\\ (\text{RKUTT1, RKUTT2})\end{matrix}$$

where h is the step size and

$$k_1 = F[x,\ s(x)], \qquad \begin{matrix}(17.6\text{-}2)\\ (\text{RKUTT1, RKUTT2})\end{matrix}$$

$$k_2 = F\left[x + \frac{h}{2},\ s(x) + \frac{h}{2}k_1\right] \qquad \begin{matrix}(17.6\text{-}3)\\ (\text{RKUTT1, RKUTT2})\end{matrix}$$

$$k_3 = F\left[x + \frac{h}{2},\ s(x) + \frac{h}{2}k_2\right] \qquad \begin{matrix}(17.6\text{-}4)\\ (\text{RKUTT1, RKUTT2})\end{matrix}$$

$$k_4 = F\left[x + h,\ s(x) + h\,k_3\right] \qquad \begin{matrix}(17.6\text{-}5)\\ (\text{RKUTT1, RKUTT2})\end{matrix}$$

This starting procedure is applied over the first three intervals of each subarc. The integration over the remainder of each subarc is accomplished by the Adams-Moulton process. This is a so-called predictor-corrector technique in which the predictor is given by

$$s_p(x+h) = s(x) + \frac{h}{24}\left[55k_1 - 59k_2 + 37k_3 - 9k_4\right] \qquad \begin{matrix}(17.6\text{-}6)\\ (\text{MADAMS})\end{matrix}$$

where, once again, h is the step size, the subscript $_p$ denotes predictor and

$$k_1 = F[x,\ s(x)] \qquad (17.6\text{-}7)\ \text{(MADAMS)}$$

$$k_2 = F[x - h,\ s(x - h)] \qquad (17.6\text{-}8)\ \text{(MADAMS)}$$

$$k_3 = F[x - 2h,\ s(x - 2h)] \qquad (17.6\text{-}9)\ \text{(MADAMS)}$$

$$k_4 = F[x - 3h,\ s(x - 3h)] \qquad (17.6\text{-}10)\ \text{(MADAMS)}$$

The corrector is given by

$$s_c(x+h) = s(x) + \frac{h}{720}\left[251\ell + 646k_1 - 264k_2 + 106k_3 - 19k_4\right] \qquad (17.6\text{-}11)\ \text{(MADAMS)}$$

where the subscript $_c$ denotes corrector and

$$\ell = F\left[x + h,\ s_p(x + h)\right] \quad \text{(Reference 9)} \qquad (17.6\text{-}12)$$

The accuracy of the corrector can be improved by means of the following optional iteration. Let the superscript ^{i} denote the iteration number. Then, as the reader may verify,

$$s_c^{(i+1)}(x+h) = s_c^{(i)}(x+h) + \frac{h}{720}\left[251\ell^{(i)} - 251\ell^{(i-1)}\right] \qquad (17.6\text{-}13)\ \text{(MADAMS)}$$

where i=1, 2, . . .,

$$\ell^{(i)} = F\left[x+h,\ s_c^{(i)}(x+h)\right] \qquad (17.6\text{-}14)$$

and

$$s_c^{(0)}(x+h) = s_p(x+h) \qquad (17.6\text{-}15)$$

We remark, in passing, that the error in the integration is proportional to h^4. Thus, if $h \cong 10^{-1}$, the error will be on the order of 10^{-4}. On the other hand, the computation time varies inversely with the size of h. Hence, cutting the step size in half will double the run time.

17.7 INTERPOLATION OF UNIVARIANT AND BI-VARIATE TABULAR FUNCTIONS

In order to expand Equation (17.1-1) into the Taylor series in Equation (17.1-5), Equation (17.1-1) had to be twice continuously differentiable with respect to Y. Since Equation (17.1-1) is functionally dependent upon both univariant and bivariate tabular functions, it is necessary to use interpolating functions for these tables that are twice continuously differentiable. Two such interpolating functions are the cubic and bicubic spline functions. The former is used for univariant data and the latter for bivariate data.

The cubic spline employed in PADS is the so-called natural cubic spline. Its derivation is given in Reference 16 and runs somewhat as follows. Let x be the independent variable and let u (not to be confused with the steering angles) be the dependent variable. Suppose the values of u are tabulated over the mesh

$$\Delta : x_0 < x_1 < \cdots < x_N \qquad (17.7\text{-}1)$$

Then, the table is given by the ordered pairs

$$(x_o, u_o),\ (x_1, u_1),\ \ldots,\ (x_N, u_N) \qquad (17.7\text{-}2)$$

Let M_i denote the value of $u''(x)$ at the i^{th} mesh point, $i = 0, 1, \ldots, N$, and let

$$h_i = x_i - x_{i-1},\ i = 1, \ldots, N \qquad (17.7\text{-}3)$$

If we suppose that the second derivative is linear over each interval in the mesh, then

$$u''(x) = M_{i-1}\frac{x_i - x}{h_i} + M_i\frac{x - x_{i-1}}{h_i}, \quad x_{i-1} \le x \le x_i \qquad \begin{matrix}(17.7\text{-}4)\\(\text{SPLINE})\end{matrix}$$

Integrating Equation (17.7-4) twice and evaluating the constants of integration yields the results

$$u(x) = M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + \left(u_{i-1} - \frac{M_{i-1}h_i^2}{6}\right)\frac{x_i - x}{h_i} + \left(u_i - \frac{M_i h_i^2}{6}\right)\frac{x - x_{i-1}}{h_i} \qquad \begin{matrix}(17.7\text{-}5)\\(\text{SPLINE})\end{matrix}$$

and

$$u'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + \frac{u_i - u_{i-1}}{h_i} - \frac{M_i - M_{i-1}}{6}h_i \qquad \begin{matrix}(17.7\text{-}6)\\(\text{SPLINE})\end{matrix}$$

for $x_{i-1} \le x \le x_i$.

The functions u(x), u'(x), and u''(x) defined by Equations (17.7-4) through (17.7-6) will be continuous at the mesh point x_i provided the quantities $M_0, M_1, \ldots, M_N$ satisfy the relationships

$$\frac{h_i}{6}M_{i-1} + \frac{h_i + h_{i+1}}{3}M_i + \frac{h_{i+1}}{6}M_{i+1} = \frac{u_{i+1} - u_i}{h_{i+1}} - \frac{u_i - u_{i-1}}{h_i} \qquad (17.7\text{-}7)$$

For i = 1, 2, . . . , N-1. At the end points we are free to close the values of M_o and M_N. We make the choice of $M_o = M_N = 0$. This implies a straight-line extrapolation outside the mesh.

Define

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \qquad (17.7\text{-}8)\ (\text{MOMENT})$$

$$\mu_i = 1 - \lambda_i \qquad (17.7\text{-}9)\ (\text{MOMENT})$$

and

$$d_i = 6\,\frac{\left[(u_{i+1} - u_i)/h_{i+1}\right] - \left[(u_i - u_{i-1})/h_i\right]}{h_i + h_{i+1}} \qquad (17.7\text{-}10)\ (\text{MOMENT})$$

for i = 1, 2, . . . , N-1. Then Equation (17.7-7) can be written as

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i \qquad (17.7\text{-}11)$$

for i = 1, 2, . . . , N-1.

Equation (17.7-11) together with the imposed end conditions on M_o and M_N yield the linear system

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mu_1 & 2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & \mu_2 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & & & \\ 0 & 0 & 0 & & 2 & \lambda_{N-2} & 0 \\ 0 & 0 & 0 & & \mu_{N-1} & 2 & \lambda_{N-1} \\ 0 & 0 & 0 & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} 0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ 0 \end{bmatrix} \qquad (17.7\text{-}12)\ (\text{MOMENT})$$

which is solved for M_0, M_1, ..., M_N. These quantities are then stored along with the table given in Equation (17.7-2) so that the functions u(x), u'(x), and u''(x) given by Equations (17.7-5), (17.7-6), and (17.7-4), respectively, can be rapidly evaluated later on.

The bicubic spline function employed in PADS is a natural extension of the cubic spline discussed above to functions of two independent variables. An excellent derivation of the bicubic spline interpolating polynomial has been given in Reference 17 and we will not attempt to duplicate it here. Suffice it to say that PADS's and Reference 17 agree in every detail except for the following:

Let the dependent variable u(x, y) be tabulated over the rectangular grid

$$\Delta_x : x_o < x_1 < \cdots < x_N$$
$$\Delta_y : y_o < y_1 < \cdots < y_M \qquad (17.7\text{-}13)$$

In other words, we are given

$$u_{ij} = u(x_i, y_j),\ i = 0, 1, \ldots, N;\ j = 0, 1, \ldots, M \qquad (17.7\text{-}14)$$

Where Reference 17 uses

$$p = u_x(x, y) \qquad (17.7\text{-}15)$$

$$q = u_y(x, y) \qquad (17.7\text{-}16)$$

and

$$s = u_{xy}(x, y) \qquad (17.7\text{-}18)$$

PADS uses

$$p = u_{xx}(x, y) \qquad (17.7\text{-}19)$$

$$q = u_{yy}(x, y) \tag{17.7-20}$$

and

$$s = u_{xxyy}(x, y) \tag{17.7-21}$$

and while Reference 17 assumes one is given

$$p_{ij} = u_x(x_i, y_j),\ i = 0, N;\ j = 0, 1, \ldots, M \tag{17.7-22}$$

$$q_{ij} = u_y(x_i, y_i),\ i = 0, 1, \ldots, N;\ j = 0, M \tag{17.7-23}$$

and

$$s_{ij} = u_{xy}(x_i, y_j),\ i = 0, N;\ j = 0, M \tag{17.7-24}$$

PADS arbitrarily takes

$$p_{ij} = u_{xx}(x_i, y_j) = 0,\ i = 0, N;\ j = 0, 1, \ldots, M \tag{17.7-25 (BLICO)}$$

$$q_{ij} = u_{yy}(x_i, y_j) = 0,\ i = 0, 1, \ldots, N;\ j = 0, M \tag{17.7-26 (BLICO)}$$

and

$$s_{ij} = u_{xxyy}(x_i y_j) = 0,\ i = 0, N;\ j = 0, M \tag{17.7-27 (BLICO)}$$

These are the bivariate analogs of the univariant end conditions $M_o = M_N = 0$.

The quantities p_{ij} at interior points of the grid are determined by univariantly spline-fitting u_{ij} along each grid line in the x direction. Similarly, the quantities q_{ij} at interior points are determined by univariantly spline-fitting u_{ij} along each grid line in the y direction. Finally, the quantities s_{ij}

at interior points are determined by spline-fitting p_{ij} along each grid line in the y direction.

The coefficients of the bicubic polynomial in PADS are given by an equation that is similar to Equation 10 of Reference 17, but in PADS the A matrix is given by

$$A(h) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/h & -h/3 & 1/h & -h/6 \\ 0 & 1/2 & 0 & 0 \\ 0 & -1/(6h) & 0 & 1/(6h) \end{bmatrix} \qquad \begin{matrix} (17.7\text{-}28) \\ (\text{BLICO}) \end{matrix}$$

REFERENCES

1. Space Shuttle Synthesis Program, General Dynamics Convair Aerospace Division Report No. GDC-DBB70-001 to -004, December 1970.

2. NASA, USAF, and USWB. U.S. Standard Atmosphere, 1962, Ed. K. S. W. Champion, U.S. Government Printing Office, Washington, D. C., 1962.

3. I. J. Kim, and J. L. Kamm. A Fast and Precise 1963 Patrick Atmosphere Analytic Model. TRW Interoffice Correspondence 5521.5-13, December 1970.

4. M. J. D. Powell. An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives. Computer Journal, Vol. 7, No. 4, 1964, pp. 303-307.

5. M. J. Box. A New Method of Constrained Optimization and a Comparison with Other Methods. Computer Journal, Vol. 8, No. 1, 1965, pp. 303-307.

6. L. D. Peterson. Trajectory Optimization by Method of Steepest Descent. Vol. 1, Formulation. AFFDL-TR-67-108, December 1967.

7. Robert E. Willwerth, Jr., Richard C. Rosenbaum, and Wong Chuck. PRESTO, Program for Rapid Earth-to-Space Trajectory Optimization. NASA CR-158, February 1965.

8. A. E. Bryson and W. F. Denham. A Steepest-Ascent Method for Solving Optimum Programming Problems. Journal of Applied Mechanics No. 29, June 1962, pp. 247-257.

9. P. Henrici. Elements of Numerical Analysis. Wiley (New York), 1964, p. 275, pp. 280-283.

10. E. Rodewig. Matrix Calculus, North Holland Publishing Co. (Amsterdam), 1956, pp. 110-114.

11. R. Courant. Differential and Integral Calculus, Vol. II, John Wiley & Sons, N. Y., 1967.

12. L. E. Elsgolc. Calculus of Variations, Addison Wesley (Reading, Mass.), 1962, p. 27.

13. G. Leitmann. Variational Problems with Bounded Control Variables, Optimization Techniques, Ed. G. Leitmann, Academic Press (New York), 1962, pp. 202-203.

14. A. E. Bryson, W. F. Denham, and S. E. Dreyfus. Optimal Programming Problems with Inequality Constraints. AIAA Journal, Vol. 1, No. 11, November 1963.

15. L. B. Rall. Computational Solution of Nonlinear Operation Equation, Wiley (New York), 1969, p. 190.

16. J. H. Ahlberg, E. N. Nilson, and J. L. Walek. The Theory of Splines and Their Applications. Academic Press (New York), 1967, pp. 9-12.

17. C. de Boor. Bicubic Spline Interpolation, J. Math. and Phys. Vol. 41, 1962, pp. 212-219.